REST API USER GUIDE

VERSION 3.18.0

UltraDNS REST API User Guide

This guide provides a detailed explanation of the REST API calls and methods available, along with JSON response examples.

neustar

If you have comments or suggestions for this guide, please contact the Neustar Support team at +1 (844) NSR-CUST, 1-844-677-2878, +1 571-434-6700 or www.support.neustar.

For additional Neustar Support and Resources, the following references may be useful:

- https://www.neustar.biz/ - Home page for the "About Neustar" experience.
- https://portal.ultradns.com/ - The UltraDNS Managed Services Portal for users.
- https://portal.ultradns.com/support.jsp - The Support page on the UltraDNS Managed Services Portal.
    - Apex Alias Tech Note
    - Traffic Management IPs for Probing
    - Important Updates to Zone Transfer IP Addresses
    - DNSSEC Quick Start Guide

Please review the *Document Revisions* to see any updates or release note items that impact content in this document.

Table of Contents

List of Figures and Tables

List of Code Examples

Introduction

This document details the Neustar UltraDNS REST API. This API allows you to:

- Create and test new API calls against a test environment that mimics your production setup.
- Use REST requests to remotely manage objects in the Neustar UltraDNS database.
- Provide an alternative to the Neustar UltraDNS Managed Services Portal (UltraDNS Portal).

URLs

Use the following base URLs for running REST API calls against the appropriate UltraDNS environment:

- REST API customer test environment for configuration information and changes: https://test-api.ultradns.com
- Production environment for configuration information and changes: https://api.ultradns.com
- Production environment for using the Reporting API: https://api.ultradns.com/reports

All of the URI constructs provided in this document use the Production URL. However, feel free to test any of your calls against the customer test environment URL to be certain the calls perform the actions you need them to. There are some limitations to the test environment. Contact Neustar customer support for information on the Customer Test Environment.

Calling the APIs

The UltraDNS APIs can accept requests and return responses in both XML and JSON formats. **The default response format is JSON unless otherwise specified (even if the request was sent in XML)**. **While XML is supported, JSON is the preferred format, and all of the examples provided in this document are in JSON**.

Controlling the format of the request and response is done by supplying the "Content-type" and "Accept" HTTP headers respectively, specifying **application/xml** or **application/json** for the value in either header (or both). Keep in mind that you do not have to specify JSON for a response.

You also have the option of using the JSON PATCH format for updates. Use the PATCH HTTP method and supply **application/json-patch+json** for the value in the "Content-type" HTTP header. For more information, see *Making Updates via JSON PATCH Format* on page 10.

UltraDNSAPI Versioning

The REST API has undergone a change in how API calls are made. UltraDNS API Versioning removes the requirement for you to add the /v1/ or /v2/ parameter in the URI when making an API call. The API call will automically use the most recent version in production.

Also, all API calls now point to a new endpoint, as noted above in the *URLs* section. This change provides greater consistency across the API calls available, as well removing the need for an "Authorization Token" for API calls, and an "Authentication Token" for Reporter API calls.

The previous method of using the full endpoint (including the version) will still work as expected, but we recommend using the new method. If you are trying to run an API call with a specific version, you must specify that version in the call. Otherwise, by default, not providing a version will default to the latest version.

There are several API calls that will return different details based upon the version (if provided) in the API call. The following table details the API calls that still return different results based upon the version used.

Table 1 API Versioning Updates and Changes

API Call	Description
Directional - TTL Update	Updating the TTL of a Directional Pool at the record level cannot be completed when using /v1/ in the API call.
Deprecated ISO Codes	Using /v1/ for a Directional API call will return different Geo-ISO details than the default (/v2) method will.
Batch API	Batch API calls can ONLY be run when using /v1/ in the API call.
Batch Query API	Batch Query API calls can ONLY be run when using /v1/ in the API call.
Reporting APIs	Report APIs no longer require the "Authentication" Token to be run. All API calls can be run using the original REST API *Authorization* Token.
Response Link Headers	When using the Response Link Headers to retrieve additional report results, the link header can ONLY be run when using the /v1/ in the API call.

Data Transfer Objects (DTOs)

Data Transfer Object (DTO) is simply another term for data structure. In this document, DTOs are the information either sent or returned for an API call, having a particular structure and containing particular information.

Each value in a DTO can be a single value of a specified *type* (Boolean, String, Integer, etc.), a series of comma-separated *values* (where permitted), or the value may be comprised of other *DTO data* and structure.

For example, if you want to create a new Primary zone, the API call to do so requires the inclusion of the **Zone Create DTO** which contains the following two fields:

- properties – Consists of the *Zone Properties DTO*.
- primaryCreateInfo – Consists of the *Primary Zone DTO*.

In turn, the Primary Zone DTO consists of several individual values and can include subsequent DTOs such as the *Restrict IP DTO* or the *Notify Address DTO*.

This document uses cross reference links, like the orange italicized ones shown above, for easy navigation to the DTO information required or returned from each call.

Responses to API Calls

All operations return a response, and all responses have a response code (HTTP Status Code). The code number returned depends on the kind of operation you sent, (Get, Put, Delete) and the status of the operation (Created, Successful, Failed, or Pending).

Successful Response Codes are returned as follows:

- Status Code 200 is typically returned for a request (GET) or modification (PUT, PATCH) of information and notes the call was Successful. If the call was a GET, you should receive a DTO containing the information you requested.
- Status Code 201 is typically returned for a POST call and indicates that the object was created.
- Status Code 202 is returned if the request has been accepted, but has yet to be completed from when the response was sent (status of Pending). These responses also include an **X-Task-ID header**.
- Status Code 204 is returned for DELETE calls, indicating the deletion was successful and there is no content to return. There is no body content presented for these responses.

If an error condition occurs, you will you receive a 400 or 500 series (4xx or 5xx) HTTP Status Code along with an HTTP body containing a specific UltraDNS error code and a description of the error. For example:

```
[
  {
     "errorCode":1801,
     "errorMessage": "Zone does not exist in the system."
  }
]
```

For detailed database transactions, a system error message is received, with a 9999 Error Code. For example:

```
[
  {
     "errorCode": 9999,
     "errorMessage": "Transaction is already completed - do not call
                      commit or rollback more than once per
                      transaction."
  }
]
```

A complete list of possible error codes and the messages that could be returned is available in the REST API Error Code Guide.

The Status Response DTO, shown below, is simply a message in the body of the response and currently returns Successful, Pending, or Created.

Table 2 Status Response DTO

Attribute	Description	Type
Message	Contains any message from the server about the result of your request.	String.

429 Error Response

Error 429 (Too Many Requests)

- This response is issued when too many requests are received from the same customer and/or IP address. The REST API monitors and restricts the frequency of incoming requests from the same customer and/or IP address for security reasons, as well as to protect the service from overloading.

Tips to Avoid the 429 Error (Too Many Requests) Response

- Re-use your authentication token multiple times, as opposed to obtaining a new authentication token every time you need to make a REST API call. (Authentication tokens currently can be re-used for up to an hour.)
- If you are still getting 429 error responses, introduce a cool down pause of 0.5-1.0 seconds between the REST API requests you make.

Two Factor Mobile Authentication

Neustar provides *Two Factor Mobile Authentication* security for the UltraDNS Portal. This is an optional tool that you can choose to enable on your accounts at no extra expense. Upon logging into the UltraDNS Portal with the Two Factor Mobile Authentication feature enabled, you will receive a six-digit Verification Code sent to your mobile device. Once the Verification Code has been provided and verified, you will have full access to the UltraDNS Portal.

Presently, Two Factor Mobile Authentication is not supported by the REST API. If any attempt is made to utilize the REST API from an account that has Two Factor Authentication currently enabled, the following error message will be returned:

```
"Two Factor Mobile Authentication security is enabled for this Login. Logging
in from <user_name> is restricted to the UltraDNS Managed Services Portal."
```

In order to utilize the REST API, the Two Factor Mobile Authentication feature will need to be disabled from the UltraDNS Portal. To learn more about Enabling and Disabling Two Factor Mobile Authentication, see the Traffic Management User Guide.

Authorization

The UltraDNS REST API uses a sub-set of OAuth 2 for authentication. This means you must know your username and password to proceed. Our form data example below assumes a username of *restapi* and password *RestAPI1*.

If you are a new user and are not able to log into the UltraDNS Portal with your username and password, check with your account owner to verify if your account has been set to only have "**API only Access**." This is a feature that only allows your username and password to give you access to the REST API.

For more information about OAuth2, see: http://apiux.com/2013/07/10/oauth-2-trumps-basic-authentication/.

https://api.ultradns.com/authorization/token

POST the request with the following form data inputs in the request body:

- grant_type = `password`
- username = `restapi`
- password = `RestAPI1`

You will receive two tokens in the response:

- The `accessToken`, used to provide your identity on subsequent REST API calls.
- The `refreshToken`, used to obtain a new access token after the previous one expires. The refresh token allows you to get a new access token without sending your username and password.

The response also contains an `expiresIn` value, which tells you the number of seconds until the accessToken expires.

The screenshot below shows the URL, the completed form data, the call type (POST), and the response body containing the tokens and the expiration.

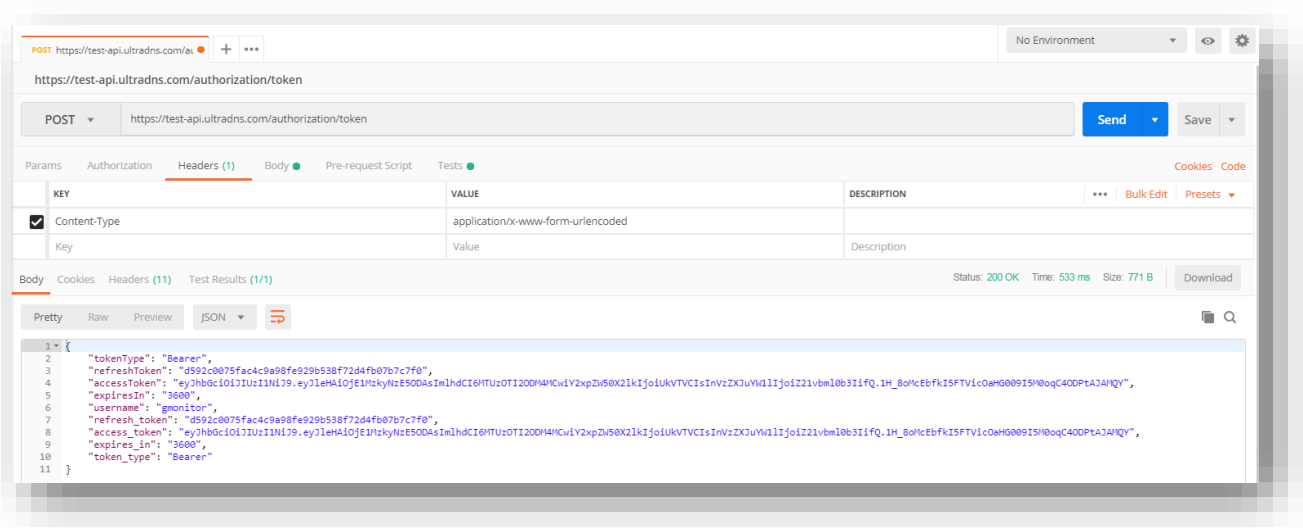

Figure 1 Obtaining access and refresh tokens

We use the Postman REST Client to provide example screenshots in this document. Postman is a freely-available REST client that allows you to save and organize frequently-used queries for later use. It can be obtained at http://www.getpostman.com/ .

Once you have an accessToken, use it in the request headers to provide authorization for subsequent requests:

```
Authorization: Bearer <token>
```

The screenshot below shows a request header with the accessToken being used for authorization.

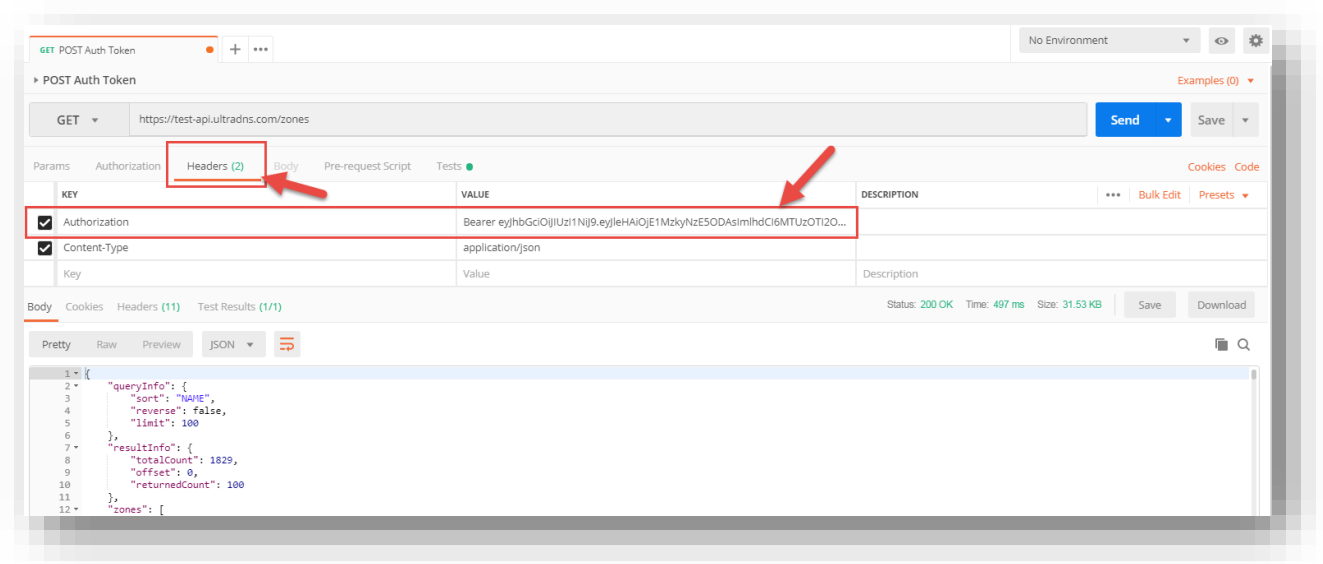

Figure 2 Adding accessToken to Request Header

When your access token expires, use the refresh token to acquire a new access token. Alternatively, you can use your login credentials to generate a new access token and new refresh token, if needed (see previous page for instructions).

To use the refresh token:

https://api.ultradns.comauthorization/token

POST the request with following credentials as form data:

- grant_type = `refresh_token`
- refresh_token = `<Refresh Token>`

A Note About Refresh Tokens:

- The refresh token expires after a single use, but has no time limit.
- Only one refresh token is valid at a time.
- If you use your username and password to acquire a new access token and refresh token, the old refresh token will automatically become expired.

The figure below shows a completed request header that uses the refreshToken to obtain a new accessToken.

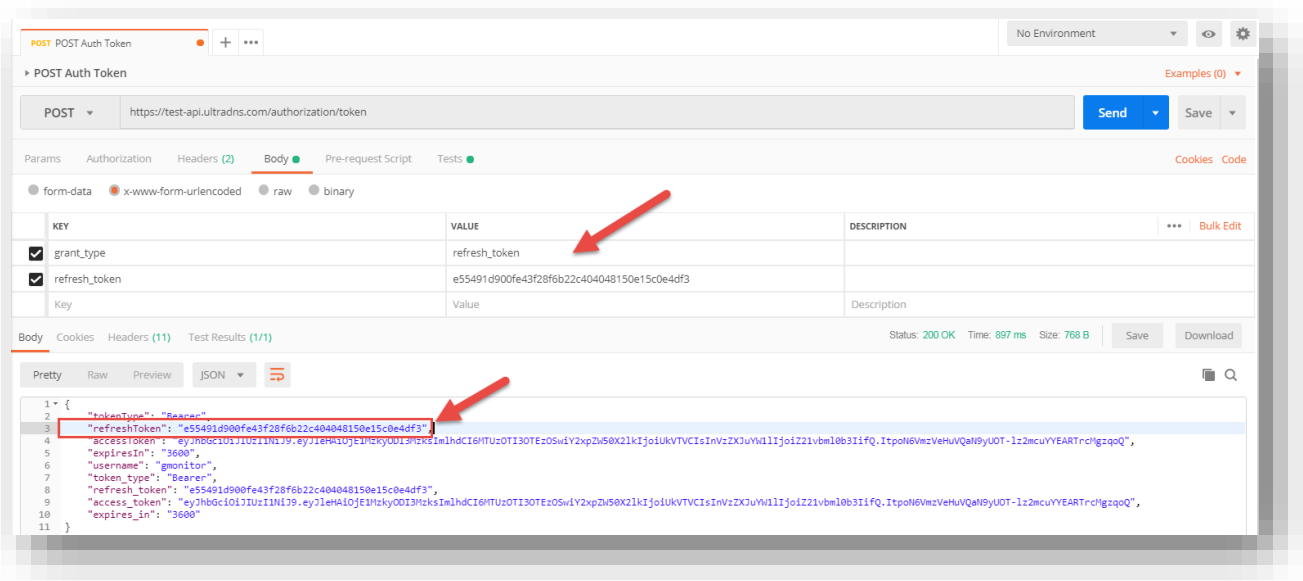

Figure 3 Using the refreshToken

Making Updates via JSON PATCH Format

UltraDNS APIs can create, modify, and return responses in both XML and JSON formats using a DTO that contains the data fields for an entity. However, there are limitations to making updates via this approach. Using PUT requires you to specify all attributes even if you are only changing one attribute. Using the current PATCH method allows you to update only the fields you need, but does not allow you to modify a value in an array, or remove an attribute or array entry.

In order to work around these limitations, the UltraDNS REST API supports the use of the JSON PATCH standardized format (defined in RFC 6902) for specifying entity updates.

JSON PATCH calls allow you to specify multiple types of updates to a single entity in the system. For example, you can use a single call to update a NotifyAddress for a zone, add a new address to the list, and remove one you no longer need.

Currently, the UltraDNS REST API allows for JSON PATCH formatted calls to:

- *Partially Update a Zone*, including Primary, Secondary and Alias
- *Partially Update Web Forward*
- *Partially Update an RRSet* – for standard RRSets, Resource Distribution (RD) pools, SiteBacker/Traffic Controller pools, and Directional pools.
- *Partially Update a Probe* – for all pool probe types.
- *Directional Pools API* – various calls for Directional Pools will allow for the usage of the JSON PATCH (each call will identify if you can use JSON PATCH)

The above links will take you to the section of this guide that contains partial update call information for each of the objects listed.

JSON PATCH requests are sent to the same endpoint as PUT or PATCH updates. However, to indicate that you are sending the request in JSON PATCH format, you must:

- Use the PATCH HTTP method.
- Supply **application/json-patch+json** for the value in the "Content-Type" HTTP header.
- Include a JSON array in the body of the request that contains one or more JSON PATCH DTOs (defined below).

A Note About JSON PATCH:

There is difference between PATCH and JSON PATCH. This difference is communicated to REST API by providing header **Content-Type: application/json-patch+json** for JSON PATCH versus **Content-Type: application/json** for regular formats like PATCH, PUT, POST, DELETE, and GET.
As a consequence, the whole batch should contain only JSON PATCH bodies, or only regular bodies.

JSON PATCH DTO

When using the JSON PATCH format you must provide a JSON PATCH DTO. The JSON PATCH DTO identifies the type of update you want to make, which attribute you want to modify, and the new value for that attribute. As with all JSON PATCH targets, the first item is numbered

0, the second is numbered 1 and so on. The JSON PATCH is constructed as shown in the below table.

Table 3 JSON PATCH DTO

Field	Description	Type
op	Patch operation type you want to perform. Valid values are: ▪ add ▪ replace ▪ remove ▪ move	String.
path	A JSON pointer that identifies the target (JSON target) on which you want to perform the provided operation. The path should be RFC-6901 compliant.	String.
value	The value you want to apply to the JSON target provided. The value is ignored when the op is set to "remove"	Object.
from	The existing value or path that needs to be changed or moved. This field is used when the “move” operation type is used.	String.

JSON PATCH Examples

The following example shows an array of JSONPATCH DTOs for updating a Primary zone. Notice that the zone is being updated using multiple operations for the different values for the zone.

JSON PATCH Example: Update Primary Zone information

```
[
  {
    "op": "replace",
    "path": "/restrictIpList/0/endIP",
    "value": "7.7.7.7"
  },
 {
    "op": "add",
    "path": "/restrictIpList/1",
    "value": {"startIP": "1.1.1.1", "endIP": "2.2.2.2"}
  },
  {
    "op": "remove",
    "path": "/restrictIpList/3"
  }
]
```

In the above example, the JSON PATCH performs the following changes to the Primary zone (identified in the PATCH call URI):

- The endIP address for the first-listed restrictIP is replaced by 7.7.7.7
- The restrictIP list has a new entry, an IP range 1.1.1.1 to 2.2.2.2. This new IP range will be shown second in a list of restrictIPs for this zone.

- The fourth-listed restrictIP entry is removed from the zone.

When updating Primary Zone Information by performing a **POST** (create) or a **PUT/PATCH** (update/partial update) call, the "restrictIPList" call is case sensitive (specificially the "IP" aspect). For these instances, the "IP" needs to be uppercase.

When performing a **GET** call, or when using **JSON-PATCH** to **PATCH** a zone, the "Ip" section of restrictIpList needs to remain as lower case ("Ip").

This next example displays the JSON PATCH operation for "move." This operation allows a record's position to be moved from one position to another within a pool. Currently, this operation is only valid for **Resource Distribution (RD)** pools, **Simple Load Balancing (SLB)** pools, and **SiteBacker / Traffic Controller (SB/TC)** pools, as these are the pools that contain multiple rdata values.
In regards to the SiteBacker and Traffic Controller pools with priority settings, once a record is moved to another position in the pool, the priority setting will be updated accordingly for the change.

JSON PATCH Example: Move operation

```
[
   {
    "op": "move",
    "path": "/rdata/2",
    "from": "/rdata/1"
   }
]
```

In the above example, the rdata is moving from position 1 to position 2 within the pool.

The example below shows a JSON PATCH DTO for updating a Secondary Zone.

JSON PATCH Example: Update Secondary Zone information

```
[
    {
     "op": "add",
     "path": "/primaryNameServers/nameServerIpList/nameServerIp1/ip",
     "value": "2.2.20.8"
    }
]
```

In this example, the IP address for the first Primary Name Server for the secondary zone has been set to 2.2.20.8. In the *Secondary Zone DTO* for this zone, this is the value that will now appear for the **nameServerIp1** attribute.

JSON PATCH format for partial updates is also supported for RRSets. However, resource record types that have multiple values within an rdata entry (such as MX, NS, or SOA) present a challenge to the standard JSON Pointer format. The standard provides no way to refer to a particular value within a single rdata entry.

To accommodate this limitation, the UltraDNS REST API provides a special case for rdata entries, allowing both the record *and* the value for that record to be optionally indexed like a list. The first target number in the path identifies the particular record in the set you want to update; the second target number in the path identifies the value for the record you want to update. If you do not list this second target number, it is assumed that you are updating all values in the specified target record.

The following example shows a JSON PATCH call to update multiple MX records in a set, demonstrating the ability to index into an rdata record.

JSON PATCH Example: Update MX records in a set

```
[
  {
    "op": "replace",
    "path": "/rdata/0/1",
    "value": "new.mail.server.biz."
  },
  {
    "op": "replace",
    "path": "/rdata/1",
    "value": "30 new3.mail.server.biz."
  }
]
```

In the above example, the first portion of the call updates only the mail server name for the first MX record in the set. The second portion of the call replaces all values (both the priority and the mail server) in the second MX record.

Performing a GET call for a resource provides the JSON structure on which the JSON-PATCH operates. The fields in the returned resource are the fields that will be specified in the path of the JSON-PATCH.

The GET calls for JSON PATCH supported items can be found on the following pages:

- *Get Zone Metadata* on page 19.
- *List all RRSets of a Type for an Owner* on page 69 (both Owner and Type are required for the partial update RRSet call).
- *Get a Probe* on page 154 for all pool probe types).
- *Get Web Forwards* on page 230.

Get the Status

The Get Status call provides a simple way to determine if the REST API is running, and to make sure you are connecting to the UltraDNS Portal with proper authorization.

Method and URI:

```
GET https://api.ultradns.com/status
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Get Status DTO* containing the status response message.

JSON Example: Status

```
{
   "message": "Good"
}
```

Errors: An error is returned under the following conditions:

- None

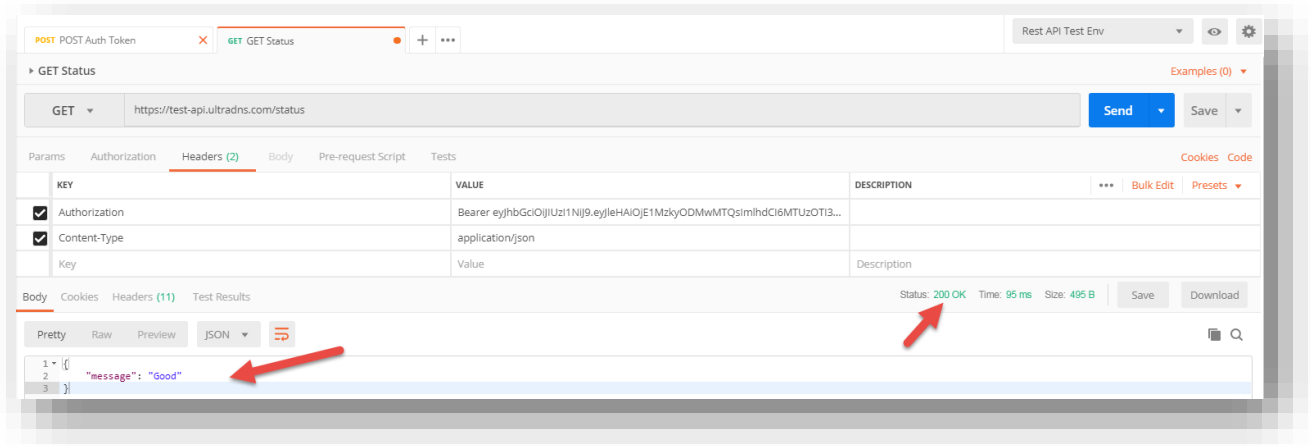

Figure 4 Get Status with Status Response Message

Get Status DTO

The following table shows the structure of the DTO returned by a request for status.

Table 4 Get Status DTO

Attribute	Description	Type
message	Contains any message from the server about the result of your request.	String.

Get the API Version

The API Version call provides the version of the REST API currently in production.

This call does not require an Authorization header to be specified, which allows it to be used to verify that there are no networking issues between a client and the REST API server.

Method and URI:

```
GET https://api.ultradns.com/version
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Version DTO* in the body content.

Errors: An error is returned under the following conditions:

- None

Version DTO

This is the structure returned by a request for the API version.

Table 5 Version DTO

Field	Description	Type
version	Contains the version of the server. If the server cannot determine its version, it will return the string "Unknown". The format of the version string is: *Major.Minor.BugFix-buildId* (example: 1.9.0-20140403224104.2beec3b)	String.

JSON Example: Version

```
{
   "version": "1.9.0-20140403224104.2beec3b"
}
```

Email Notifications

You are able to update the notification email address that is on file via the RESTful interface, and using the *Making Updates via JSON PATCH Format*.

Email notifications commonly are sent for Zone Transfer Notification issues (failures or threshold settings being exceeded), DDOS Notifications, SiteBacker and Traffic Controller Probe Events, Record Events, and Scheduled Events.

Update Notification Email Address

Method and URI:

```
PATCH https://api.ultradns.com/zones/{secondaryZoneName}
```

JSON Example: Update Notification Email Address

```
{
"secondaryCreateInfo":
  {
    "notificationEmailAddress":"<updated email address>"
  }
}
```

Zone API

A DNS Zone is a portion of a DNS Domain separated for administrative control. You can think of it as a container for individual DNS Resource Records. Zones (domains) are the basic building blocks of DNS. UltraDNS defines three classes of zone types: *Primary*, *Secondary*, and *Alias*.

- A **Primary Zone** is the master copy of the zone data. UltraDNS manages Primary zones. They may include advanced features like pools.
- A **Secondary Zone** is a copy of the primary zone, and is owned and controlled by a nameserver outside the UltraDNS system. UltraDNS retrieves a copy via zone transfer of the zone from the remote nameserver. Secondary zones are read-only (except for requests for transfer) and cannot contain advanced UltraDNS features.
- An **Alias Zone** is a virtual copy of a Primary zone; it's basically the primary zone under a different zone name. They are read-only, but contain all of the advanced features of the primary zones they alias.

This chapter provides details on the Zone API calls available for use, as well as detailed **Zone DTO** (Data Transfer Object) information. When DTOs are required in the body of the call, or are returned as a response, cross reference links are provided to the specific table containing the details of DTO contents.

To escape forward slashes in zone names (for example, a reverse zone with the name 0/24.50.156.193.in-addr.arpa), use `%2F`.
In our example URIs, to specify the reverse zone noted above:
`https://api.ultradns.com/zones/0%2F24.50.156.193.in-addr.arpa`.

Create a Zone

The Create Zone API allows you to create a Primary, Secondary, or Alias Zone, and furthermore, allows you to create a Zone "from scratch" by copying another zone via an uploaded file or by Zone transfer. The JSON examples provided below give a sample of each type of zone create call.

Create a Zone is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones
```

Parameters: None

Body: Must include a *Zone Create DTO*.

The Zone Create DTO requires the inclusion of a Zone Properties DTO. Depending on the type of Zone you are creating, you will also require a Primary Zone DTO, a Secondary Zone DTO or an Alias Zone DTO.

For DTO reference, see the following tables:

- *Table 7 Zone Properties DTO* on page 34
- *Table 8 Zone Create DTO* on page 34

- *Table 9 Primary Zone DTO* on page 37
- *Table 13 Secondary Zone DTO* on page 40
- *Table 14 Alias Zone DTO* on page 41

Response: If task completes, Status Code 201 is returned with an appropriate message in the response body.

- If creation happens in the background, a Status Code 202 is returned with a status response message of “Pending” along with an X-Task-Id header in body content.

Errors: An error is returned under the following conditions:

- If {zoneName} already exists.
- If {zoneName} is not valid.
- If you don't have permission to create zones.
- If creating a Primary Zone via copy, if creating an Alias, or if the original zone is not a Primary zone.

JSON Example: Create a Zone

```
{
   "properties":{
      "name":"changecommentdemo.com",
      "accountName":"demoaccount",
      "type":"PRIMARY"
   },
   "primaryCreateInfo":{
      "forceImport":true,
      "createType":"NEW"
   },
   "changeComment":"Create zone as agreed"
}
```

Delete a Zone

The Delete Zone API allows you to delete any zone you have the proper authority to delete. You cannot delete a primary zone if it has an Alias zone.

Delete Zone call is generated as follows:

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls.	String.

Response: If delete happens immediately, Status Code 204 returned with no body content.

- If delete happens in the background, a Status Code 202 is returned with a status response message of Pending, along with an X-Task-Id header in body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to delete {zoneName}.
- If {zoneName} does not exist.

JSON Example: Delete Zone with Change Comment

```
{
      "changeComment": "Deleting Zone as agreed"
}
```

Get Zone Metadata

The Get Zone Metadata call returns Zone information for the specified {zoneName} in the form of a *Zone DTO*. This DTO can in turn be used for other calls as needed.

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with *Zone DTOs* in the body content. Example responses for different zone types are shown below.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to read {zoneName}.

JSON Example: Responses to Primary Zone – Get Metadata

```
{
   "properties": {
      "name": "primary-example.com.",
      "accountName": "example",
      "owner": "example",
      "type": "PRIMARY",
```

```
      "recordCount": 3,
      "dnssecStatus": "UNSIGNED",
      "lastModifiedDateTime": "2014-07-01T22:13Z"
   },
   "registrarInfo": {
      "registrar": "Generic Domain Name Registrar",
      "whoisExpiration": "2015-01-01 00:00:00",
      "nameServers": {
         "ok": ["PDNS1.ULTRADNS.NET", "PDNS2.ULTRADNS.NET"],
      }
   },
   "restrictIpList": [
      {
         "startIP": "10.20.30.40",
         "endIP": "20.20.20.20",
         "comment": "Comment"
      }
   ],
   "tsig": {
      "tsigKeyName":"Key",
      "tsigKeyValue": "This would be a hash if it was real",
      "description": "TSIG for primary-example.com",
      "tsigAlgorithm": "hmac-sha256"
   },
   "notifyAddresses": [
      {
         "notifyAddress": "2.4.5.6",
         "description": "East Coast Server"
      },
      {
         "notifyAddress": "5.6.7.8",
         "description": "West Coast Server"
      }
   ]
}
```

JSON Example: Secondary Zone – Get Metadata

```
{
   "properties": {
      "name": "secondary-example.com.",
      "accountName": "example",
      "owner": "example",
      "type": "SECONDARY",
      "recordCount": 3,
      "dnssecStatus": "UNSIGNED",
      "lastModifiedDateTime": "2014-07-01T22:13Z"
   },
   "primaryNameServers": {
      "nameServerIpList": {
         "nameServerIp1": {
            "ip": "1.2.3.4",
            "tsigKey": "key1",
            "tsigKeyValue": "value1"
         },
         "nameServerIp2": {
```

```
            "ip": "2.4.5.6",
            "tsigKey": "key2",
            "tsigKeyValue": "value2"
         },
         "nameServerIp3": {
            "ip": "3.4.5.6",
            "tsigKey": "key3",
            "tsigKeyValue": "value3"
         }
      }
   }
         "transferStatusDetails": {
         "lastRefresh": "06/13/18 06:07:45 AM GMT",
         "nextRefresh": "06/13/18 07:07:45 AM GMT",
         "lastRefreshStatus": "FAILED",
         "lastRefreshStatusMessage": "Failed to transfer zone 'secondary-
           example.com.' from host: 54.209.41.82; reason:
           java.net.SocketTimeoutException"
      }
   }
}
```

JSON Example: Alias Zone – Get Metadata

```
{
   "properties": {
      "name": "alias-example.com.",
      "accountName": "example",
      "owner": "example",
      "type": "ALIAS",
      "recordCount": 3,
      "dnssecStatus": "UNSIGNED",
      "lastModifiedDateTime": "2014-07-01T22:13Z"
   },
   "originalZoneName": "example.com."
}
```

List Metadata for Zones

The List Metadata for Zones call differs from the Get Zone Metadata call in that it provides a summary list of all zones (or all zones of a specified type), rather than metadata for a particular zone. The List Metadata for zones call is a GET call and is generated as follows:

Method and URI:

```
GET https://api.ultradns.com/zones/
```

Parameters: Parameters are listed in the following table:

Table 6 Parameters for get metadata for zones

Parameter	Description	Type
q	The query used to construct the list. Query operators are: ▪ **name** – Name of the zone (allowing for partial string matches).	String.

Parameter	Description	Type
	▪ **zone_type** – Returns zones of an identified type. If not specified, all zone types are returned. Valid values are ALIAS, PRIMARY, or SECONDARY. ▪ **zone_status** – Returns zones with the identified status. Active zones are returned if not specified. Valid values are ACTIVE, SUSPENDED, or ALL. ▪ **dnssec_status** – Returns zones based upon the dnssec status. Valid values are SIGNED or UNSIGNED. If not specified, both types of zones will be returned. ▪ **account_name** – Returns the zones based upon the account. If not specified, zones of all of the accounts that the user has access to will be returned. ○ If the account name has space characters in it, the space characers need to be replaced with "**%20**." For example, account "**test account**" will need to be "**test%20account**."	
offset	The position in the list for the first returned element (0 based). Default is "0."	Integer.
limit	The maximum number of rows requested. Default is 100.	Integer.
sort	The sort column used to order the list. The valid values are: ▪ NAME (default sort column) ▪ ACCOUNT_NAME ▪ ZONE_TYPE	String.
reverse	List is sorted in Ascending order by default, with the parameter value being **false**. Enter **true** to sort the list in Descending order by the sort column specified (or by Name if no sort value is entered).	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Zone List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to read zones.

JSON Example: Zone List

```
{
   "queryInfo": {
      "q": "",
      "sort": "NAME",
      "reverse": false,
      "limit": 100
   },
   "resultInfo": {
      "totalCount": 3,
      "offset": 0,
      "returnedCount": 3
   }
   "zones": [
```

```
    {
       "properties": {
          "name": "alias-example.com.",
          "accountName": "example",
          "owner": "example",
          "type": "ALIAS",
          "recordCount": 3,
          "dnssecStatus": "UNSIGNED",
          "lastModifiedDateTime": "2014-07-01T22:13Z"
       },
       "originalZoneName': "example.com."
    },
    {
       "properties": {
          "name": "primary-example.com.",
          "accountName": "example",
          "owner": "example",
          "type": "PRIMARY",
          "recordCount": 3,
          "dnssecStatus": "UNSIGNED",
          "lastModifiedDateTime": "2014-07-01T22:13Z"
       },
       "registrarInfo": {
          "registrar": "Generic Domain Name Registrar",
          "whoisExpiration": "2015-01-01 00:00:00",
          "nameServers": {
             "ok": ["PDNS1.ULTRADNS.NET", "PDNS2.ULTRADNS.NET"],
          }
       },
       "restrictIpList": [
          {
             "startIP": "10.20.30.40",
             "endIP": "20.20.20.20",
             "comment": "Comment"
          }
       ],
       "tsig": {
          "tsigKeyName":"Key",
          "tsigKeyValue": "This would be a hash if it was real",
          "description": "TSIG for primary-example.com",
          "tsigAlgorithm": "hmac-sha256"
       },
       "notifyAddresses": [
          {
             "notifyAddress": "2.4.5.6",
             "description": "East Coast Server"
          },
          {
             "notifyAddress": "5.6.7.8",
             "description": "West Coast Server"
          }
       ]
    },
    {
       "properties": {
          "name": "secondary-example.com.",
```

```
            "accountName": "example",
            "owner": "example",
            "type": "SECONDARY",
            "recordCount": 3,
            "dnssecStatus": "UNSIGNED",
            "lastModifiedDateTime": "2014-07-01T22:13Z"
         },
         "primaryNameServers": {
            "nameServerIpList": {
               "nameServerIp1": {
                  "ip": "1.2.3.4",
                  "tsigKey": "key1",
                  "tsigKeyValue": "value1"
               },
               "nameServerIp2": {
                  "ip": "2.4.5.6",
                  "tsigKey": "key2",
                  "tsigKeyValue": "value2"
               },
               "nameServerIp3": {
                  "ip": "3.4.5.6",
                  "tsigKey": "key3",
                  "tsigKeyValue": "value3"
               }
            }
         }
      }
   ],
}
```


Convert a Zone

The Convert Zone call converts a Secondary Zone into a Primary Zone. The Convert Zone call is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/convert
```

Parameters: None

Body: Optionally, can include the *Zone Create DTO* and the use of the changeComment field. If providing DTO fields, the "Content-Type: application/json" header is required.

Response: If conversion completes, Status Code 201 is returned with an appropriate status message in the response body.

- If conversion happens in the background, a Status Code 202 is returned along with an X-Task-ID header and a status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have write permission for {zoneName}.

- If {zoneName} is not a secondary zone.

JSON Example: Convert Zone with Change Comment

```
{
      "changeComment": "Converting zone 12/05/2020"
}
```

Unalias a Zone

Unaliasing a zone is the process of converting an Alias Zone into a Primary Zone. When you unalias a zone, the following changes happen:

- All of the data and zone configuration information is copied from the Primary to the Alias.
- The Alias is converted into a Primary zone.
- Any correlation between the original Primary and new Primary (formerly the Alias) is removed; the two are now wholly separate Primary zones.

The Unalias call is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/unalias
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to unalias {zoneName}.
- If {zoneName} is a not an alias zone.

JSON Example: Unalias Zone with Change Comment

```
{
      "changeComment": "Unalias this zone. No longer required."
}
```

Suspend a Zone

Suspending a zone allows you to temporarily stop serving data for a zone without deleting that zone. When you suspend a zone, the following changes happen:

- The zone cannot be updated via a PUT or PATCH
- Performing a GET will still return zone data

The Suspend call is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/suspend
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to suspend {zoneName}.

JSON Example: Suspend Zone with Change Comment

```
{
      "changeComment": "Suspending Zone until 2/20/2021"
}
```

UnSuspend a Zone

Un-Suspending a zone is the process of re-enabling a suspended zone. When you unsuspend a zone, the following changes happen:

- The zone can be updated via PUT or PATCH

The UnSuspend call is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/unsuspend
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to unsuspend {zoneName}.
- If the Zone is not currently suspended.

Update a Zone

The Update Zone call allows you to update certain aspects of either a Primary or a Secondary Zone. You ***cannot*** use this call to:

- Update an Alias Zone.
- Specify Primary Name Servers for a Primary zone.
- Specify restrict IPs, TSIG, or Notify addresses for a Secondary Zone.

As this is a FULL update (replacing data) for Primary Zone updates, you must include any necessary restrict IPs, Notify addresses, or Primary Name Servers that apply. Any data not included with the update **will be deleted** from the Primary Zone. See also *Partially Update a Zone* on page 29.

Update Zone is a PUT call and is generated as follows:

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}
```

Parameters: None

Body: Must include a *Zone Create DTO*, specifically containing information as follows:

- ***To update a Primary Zone***, include only the **createPrimaryInfo** section. This section consists of a *Primary Zone DTO*, of which you only need to provide the **restrictIPList**, **tsig**, and/or **notifyAddresses** sections.

  IMPORTANT: Because this is a full update, any restrictIPs, tsig, or notifyAddresses not included *will be deleted* from the Primary Zone (unless the zone inherits the setting from the account).

- ***To update a Secondary Zone***, include only the **createSecondaryInfo** section. This section consists of a *Secondary Zone DTO*, of which you only need to provide the **primaryNameServers** section.

  IMPORTANT: Because this is a full update, any primaryNameServers not included *will be deleted* from the Secondary Zone.

Examples of the information to be provided are shown below. If additional sections are sent, they will be ignored.

For DTO reference, see the following tables:

- *Table 8 Zone Create DTO* on page 34
- *Table 9 Primary Zone DTO* on page 37
- *Table 13 Secondary Zone DTO* on page 40

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have write permission for {zoneName}.
- If zone is an Alias Zone (cannot update Alias zones).
- If the wrong kind of data is submitted for a zone (see above for information required).

JSON Example: Update Restrict IP details for Primary Zone

```
{
   "primaryCreateInfo": {
      "restrictIPList": [
         {
            "startIP": "10.20.30.40",
            "endIP": "20.20.20.20",
            "comment": "Comment"
         }
      ]
   },
```

```
   "changeComment": "Updating zone"
}
```

JSON Example: Update TSIG and Notify information for Primary Zone

```
{
   "primaryCreateInfo": {
      "tsig": {
         "tsigKeyName": "Key",
         "tsigKeyValue": "This would be a hash if it was real",
         "description": "TSIG for primary-example.com",
         "tsigAlgorithm": "hmac-sha256"
      },
      "notifyAddresses" : [
         {
            "notifyAddress": "2.4.5.6",
            "description": "East Coast Server"
         },
         {
            "notifyAddress": "5.6.7.8",
            "description": "West Coast Server"
         }
      ]
   }
}
```

JSON Example: Update Primary Name Server information for Secondary Zone

```
{
   "secondaryCreateInfo": {
      "primaryNameServers": {
         "nameServerIpList": {
            "nameServerIp1": {
               "ip": "1.2.3.4",
               "tsigKey": "key1",
               "tsigKeyValue": "value1"
            },
            "nameServerIp2": {
               "ip": "2.4.5.6",
               "tsigKey": "key2",
               "tsigKeyValue": "value2"
            },
            "nameServerIp3": {
               "ip": "3.4.5.6",
               "tsigKey": "key3",
               "tsigKeyValue": "value3"
            }
         }
      }
   }
}
```

Partially Update a Zone

The Partial Update a Zone call is used to:

- Update the restrictIPs, TSIG key, and/or Notify Address information for a Primary zone without having to explicitly list all of them. Any Restrict IPs, TSIGs or Notify Addresses not included in the call are retained on the server.
- Update the Primary Name Servers for a Secondary zone without having to explicitly list all of them. Any Primary Name Servers not included in the call are retained on the server.

Alias zones cannot be updated.

Partially Update a Zone is a PATCH or a JSON PATCH call and is generated as follows:

Method and URI:

```
PATCH https://api.ultradns.com/zones/[zoneName}
```

Parameters: None

Body: For standard XML or JSON formatted calls, the body must include a *Zone Create DTO*, specifically containing information as follows:

- ***To update a Primary Zone***, include only the **createPrimaryInfo** section. This section consists of a *Primary Zone DTO*, of which you only need to provide the **restrictIpList**, **tsig**, and/or **notifyAddresses** sections.

  Because this is a partial update, any restrictIps, tsig, or notifyAddresses not included in the call will be retained on the Primary zone.
- ***To update a Secondary Zone***, include only the **createSecondaryInfo** section. This section consists of a *Secondary Zone DTO*, of which you only need to provide the **primaryNameServers** section, or the **notificationEmailAddress.**

  Because this is a partial update, any primaryNameServers not included will be retained on the Secondary Zone. If additional sections are sent, they will be ignored.

For JSON PATCH formatted updates, the body must include a *JSON PATCH DTO*.

For DTO reference, see the following tables:

- *Table 3 JSON PATCH DTO* on page 11.
- *Table 8 Zone Create DTO* on page 34.
- *Table 9 Primary Zone DTO* on page 37.
- *Table 13 Secondary Zone DTO* on page 40.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have write permission for {zoneName}.

- If zone is an Alias Zone (Alias Zones cannot be updated).
- If the wrong kind of data is submitted for a zone (see above for information required).

JSON Example: Partially Update a Zone with Change Comment

```
{
   "primaryCreateInfo":{
      "notifyAddresses":[
         {
            "notifyAddress":"2.4.5.6",
            "description":"East Coast Server"
         },
         {
            "notifyAddress":"5.6.7.8",
            "description":"West Coast Server"
         }
      ]
   },
   "changeComment": "Add notify ips using Patch"
}
```

Request Zone Transfer

The Request Zone Transfer call sends an AXFR request through a Secondary Zone, to the primary name server in order to update the Secondary Zone with information from the Primary Zone.

The {zoneName} identified in the call should be the name of the Secondary Zone to be updated.

The Request Zone Transfer call is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/transfer
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to update {zoneName}.
- If {zoneName} does not refer to a Secondary Zone.

Export a Zone

Exporting a Zone will create a task to export the zone details into a BIND file. Once the task has been completed the BIND file can be downloaded.

Method and URI:

```
POST https://api.ultradns.com/zones/export
```

Parameters: None

Body: The body must include the zonename that is being exported.

Field	Description	Type
zoneNames	The name or names of the zones that are being exported, with or without the trailing dot. Multiple zone names must be comma separated.	String. Must be a valid zone name.

Response: If task completes, Status Code 202 Accepted is returned along with an X-Task-ID header and status message of Pending in the body content. To check the status of the export, use the *Get the Status of a Task* call.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to export {zoneName}.
- If zoneNames contains more than 250 zone names for export.

JSON Example: Export a Zone Body Example

```
{
    "zoneNames": ["name.com", "name2.com."]
}
```

JSON Example: Export Zone BIND File Details

```
;File created: 12/01/2020 16:13
;Record count: 6
$ORIGIN 00-ben-doc-ns.com.
@  86400 IN SOA udns1.ultradns.net. rajender\.aindla.neustar.biz. (
         2018062747  ;Serial
         10800    ;Refresh
         3600     ;Retry
         2592000     ;Expire
         10800    ;Minimum
      )
@  86400 IN NS udns1.ultradns.net.
@  86400 IN NS udns2.ultradns.net.
www.momandpopgas.com 600   IN A  1.1.1.1
;record belongs to the pool HealthProbeTest.com.00-ben-doc-ns.com.
HealthProbeTest.com  120   IN A  6.5.4.3
```

```
mydeadpool.com 60 IN A  2.2.2.2
```

Get the Status of a Export Zone Task

Method and URI:

```
GET https://api.ultradns.com/tasks/{taskId}
```

Parameters: Must include the specific Task ID.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Task DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If you do not have permission to read {taskId}.

Get the Results of a Task

Method and URI

```
GET https://api.ultradns.com/tasks/{taskId}/result
```

Parameters: Must include a Task ID.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Task DTO* in the body content.

The content will be returned as a downloadable file. The name of the file will be the {taskId} that was submitted with the request. The file extension and content type are set by the background task and will be appropriate to the data returned.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If you do not have permission for the task associated with the supplied {taskId}.
- If task is not yet completed.

Zone DTOs

The sections and tables below provide detailed information about the contents of the DTOs used for Zone API calls. When a DTO field consists of the contents of another DTO, a cross reference link to the associated DTO is provided. When possible, return links to the "parent" DTO are provided, along with links to the API calls that use the DTO.

Zone Properties DTO

The Zone Properties DTO holds the common metadata across all types of zones. This must be included in the *Zone Create DTO* used for *Create a Zone* call, unless it is present on an update, in which case it can be ignored.

Table 7 Zone Properties DTO

Field	Description	Type
name	Name of the zone, with trailing periods (….)	Must be a valid domain name. Required for zone creation. Ignored if present on update.
accountName	Name of the account.	String. Required for zone creation. Ignored if present on update.
type	Type of zone. Valid values are PRIMARY, SECONDARY or ALIAS.	Required for zone creation. Ignored if present on update.
owner	Name of the user that created the zone.	String. Returned in GET responses for zone information. Ignored if present on create or update.
recordCount	Number of records in the zone	Integer. Returned in GET responses for zone information. Ignored if present on create or update.
dnssecStatus	Whether or not the zone is signed with DNSSEC. Valid values are SIGNED or UNSIGNED.	Returned in GET responses for zone information. Ignored if present on create or update.
lastModifiedDateTime	The last date and time the zone was modified, represented in ISO8601 format.	Returned in GET responses for zone information. Ignored if present on create or update.

Zone Create DTO

The Zone Create DTO is the data structure used for the *Create a Zone*, and *Partially Update a Zone* API calls.

Table 8 Zone Create DTO

Field	Description	Type
properties	The name, account name, and type of zone being created.	*Zone Properties DTO*. Required for zone creation. Ignored if present on update.

Field	Description	Type
primaryCreateInfo	Metadata for a primary zone.	*Primary Zone DTO*. Required to create or update a primary zone, ignored in all other cases.
secondaryCreateInfo	Metadata for a secondary zone.	*Secondary Zone DTO*. Required to create or update a secondary zone, ignored in all other cases.
aliasCreateInfo	Metadata for an alias zone.	Alias Zone DTO. Required to create an alias zone, ignored in all other cases.
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text. Applicable for all Zone api calls. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

JSON Example: New Primary Zone

```
{
      "properties": {
         "name": "primary-example.com.",
         "accountName": "example",
         "type": "PRIMARY"
      },
      "primaryCreateInfo": {
         "forceImport": true,
         "createType": "NEW"
      },
      "changeComment": "Created as agreed"

}
```

JSON Example: New Primary Zone Copied from Another Zone

```
{
      "properties": {
         "name": "copy-example.com.",
         "accountName": "example",
         "type": "PRIMARY"
      },
      "primaryCreateInfo": {
         "forceImport": true,
         "createType": "COPY",
         "originalZoneName": "example.com."
      }
}
```

JSON Example: New Primary Uploaded from a File

```
{
```

```
      "properties": {
         "name": "example.com.",
         "accountName": "example",
         "type": "PRIMARY"
      },
      "primaryCreateInfo": {
         "forceImport": true,
         "createType": "UPLOAD"
      }
}
```

Bind Upload – TTL Behavior during Zone Creation

First Scenario

When creating a Zone via bind upload, any records with the same owner and type, but have different TTLs (in the bind file), will be created with the lowest TTL value amongst all the records of the same owner and type (in the bind file).

For example, the following three records for the owner “txtrecord” and the type TXT, along with rdata and TTLs in one bind file:

1. *txtrecorddata1 with TTL 300*
2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata3 with TTL 400*

In this scenario, all three of the above records would be created with the TTL value 300.

Second Scenario

Similarly to the previous example, if you have records with the same owner and type in a bind file, but only some of the TTLs are provided for the records in the rrset, then all of the records of the rrset will inherit the lowest TTL value (provided in the bind file).

1. *txtrecorddata1 with no TTL*
2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata3 with TTL 400*

All of the above records in the example will be given the TTL value of 400. If none of the provided records have TTL values, then the TTL value for each record will be given the default TTL value of 86400.

JSON Example: New Primary Zone via Transfer

```
{
      "properties": {
         "name": "copy-example.com.",
         "accountName": "example",
         "type": "PRIMARY"
      },
      "primaryCreateInfo": {
         "forceImport": true,
         "createType": "TRANSFER",
```

```
          "nameServer": {
             "ip": "1.2.3.4",
             "tsigKey": "key",
             "tsigKeyValue":"value"
          }
       }
}
```

Primary Zone DTO

The Primary Zone DTO contains the metadata used to create or update a Primary Zone. The *Create a Zone*, and *Partially Update a Zone* API calls use the *Zone Create DTO,* which in turn references this DTO.

Table 9 Primary Zone DTO

Field	Description	Type
forceImport	Whether or not to move existing records from zones into this new zone. Values include: ▪ **"true"** = move ▪ "**false**" = leave in existing zone (default)	Boolean. Only used for primary zone creation. If not present, defaults to "false". Ignored if present for update.
createType	Indicates the method for creating the primary zone. Values include: ▪ "NEW" ▪ "COPY" ▪ "TRANSFER" ▪ "UPLOAD"	Required for primary zone creation. Ignored if present for update.
nameServer/ip	IP address of the primary zone's name server (where the primary zone is being transferred from).	IPv4 or IPv6 address. Required if createType is "TRANSFER." Ignored if present for update.
nameServer/tsigKey	If TSIG is enabled for this name server, the name of the TSIG key.	String. Used only if createType is "TRANSFER." Required if TSIG is enabled for this name server. Ignored if present for update.
nameserver/tsigKey Value	If TSIG is enabled for this name server, the TSIG key's value.	String. Used only if createType is "TRANSFER." Required if TSIG is enabled for this name server. Ignored if present for update.

Field	Description	Type
nameserver/ tsigAlgorithm	The hash algorithm used to generate the TSIG key. Valid values are: ▪ hmac-md5 ▪ hmac-sha1 ▪ hmac-sha224 ▪ hmac-sha256 ▪ hmac-sha384 ▪ hmac-sha512	String. Used only if createType is "TRANSFER." Default is hmac-md5. Required if TSIG is enabled for this name server. Ignored if present for update.
originalZoneName	The name of the zone being copied. The existing zone must be owned by the same account as the new zone.	String. Must be a valid domain name. Required if createType is "COPY." Ignored if present for update.
restrictIPList	The list of IP ranges that are allowed to transfer primary zones out using zone transfer protocol (AXFR/IXFR).	List of *Restrict IP DTO*s. Optional for both creation and update.
tsig	The TSIG information for the primary zone.	*TSIG DTO*. Optional for both creation and update.
notifyAddresses	The addresses that are notified when updates are made to the primary zone.	List of *Notify Address DTO*s. Optional for both creation and update.
inherit	Defines whether this zone should inherit the zone transfer values from the Account, and also specifies which values to inherit. Defaults to 'ALL' if zone transfer settings on the account have been set.	Optional for both creation and update. Valid values include: ▪ ALL ▪ NONE ▪ Any combination of: ○ IP_RANGE ○ NOTIFY_IP ○ TSIG Separate multiple values with a comma, i.e., IP_RANGE, NOTIFY_IP

Restrict IP DTO

Each Restrict IP DTO holds the IP addresses that are allowed to transfer Primary Zones out using the Zone Transfer protocol (AXFR/IXFR). The Restrict IP DTO contains information used in the *Primary Zone DTO*.

The IP address information can be specified in three different formats:

1. IP Range (startIP and endIP)
2. CIDR (cidr)

3. Single IP (singleIP)

Only one format should be specified in the DTO at a time (range, CIDR or single IP).

Table 10 Restrict IP DTO

Field	Description	Type
startIP	The start of the IP range that is allowed to transfer this primary zone out using zone transfer protocol.	IPv4 or IPv6 address.
endIP	The end of the IP range that is allowed to transfer this primary zone out using zone transfer protocol.	IPv4 or IPv6 address/
cidr	The IP ranges specified in CIDR	CIDR (e.g. 1.1.1.1/30, ::10/126)
singleIP	The IP that is allowed to transfer this primary zone out using zone transfer protocol.	IPv4 or IPv6 address.
comment	A description of this range of IP addresses.	String. Optional.

TSIG DTO

The TSIG DTO holds TSIG information for the Primary Zone. The TSIG DTO contains information used in the *Primary Zone DTO*.

Table 11 Tsig DTO

Field	Description	Type
tsigKeyName	The name of the TSIG key for the zone.	String. REQUIRED.
tsigKeyValue	The value of the TSIG key for the zone.	String. REQUIRED.
description	A description of this key.	String. Optional.
tsigAlgorithm	The hash algorithm used to generate the TSIG key. Valid values are: ▪ hmac-md5 ▪ hmac-sha1 ▪ hmac-sha224 ▪ hmac-sha256 ▪ hmac-sha384 ▪ hmac-sha512	String. REQUIRED.

Notify Address DTO

Each Notify Address DTO defines an address that gets notified when there are updates to a Primary Zone. The Notify Address DTO contains information used in the *Primary Zone DTO*.

Table 12 Notify Address Detail DTO

Field	Description	Type
notifyAddress	The IP address that is notified when the primary zone is updated.	IPv4 address. REQUIRED.

Field	Description	Type
description	A description of this address.	String. Optional.

Secondary Zone DTO

The Secondary Zone DTO holds the metadata used to create or update a Secondary Zone. The *Create a* Zone, and *Partially Update a Zone* API calls use the *Zone Create DTO,* which in turn references this DTO.

This DTO is also used to return the Primary Name Servers for a Secondary Zone when the *Get Zone Metadata* call is used.

Table 13 Secondary Zone DTO

Field	Description	Type
primaryNameServers	The primary name servers of the source zone for the secondary zone.	*Name Server IP List DTO*. Required for creating or updating a secondary zone. Ignored in all other cases.
notificationEmailAddress	The Notification Email for a secondary zone.	String. Optional.

JSON Example: New Secondary Zone

```
{
    "properties": {
       "name": "secondary-example.com.",
       "accountName": "example",
       "type": "SECONDARY"
    },
    "secondaryCreateInfo": {
        "primaryNameServers": {
          "nameServerIpList": {
          "nameServerIp1": {
          "ip": "1.2.3.4",
          "tsigKey": "key1",
          "tsigKeyValue": "value1"
      }
       }
   },
   "notificationEmailAddress": "<email_address>"
    }
}
```

Alias Zone DTO

The Alias Zone DTO holds the metadata used for creating an Alias Zone. The *Create a Zone* API call uses the *Zone Create DTO* which in turn references this DTO.

Table 14 Alias Zone DTO

Field	Description	Type
originalZoneName	The name of the zone being aliased. The existing zone must be owned by the same account as the new zone.	Must be a valid domain name. Required for alias during creation.

JSON Example: New Alias Zone

```
{
   "properties": {
      "name": "alias-example.com.",
      "accountName": "example",
      "type": "ALIAS"
   },
   "aliasCreateInfo": {
      "originalZoneName": "example.com."
   },
   "changeComment":"Create an alias zone"

}
```

Name Server IP List DTO

The Name Server IP List DTO lists the Primary Name Servers for a Secondary Zone. It is referenced by the *Secondary Zone DTO* which is used for the *Create a Zone*, *Partially Update a Zone*, and *Get Zone Metadata* API calls.

Table 15 Name Server IP List DTO

Field	Description	Type
nameServerIpList /nameServerIP1/ip	The IP address of the primary name server for the source zone.	IPv4 or IPv6 address. Required for creation or update of a secondary zone.
nameServerIpList /nameServerIP1/tsig Key	If TSIG is enabled for this name server, the name of the TSIG key.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList /nameServerIP1 /tsigKeyValue	If TSIG is enabled for this name server, the TSIG key's value.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.

Field	Description	Type
nameServerIpList /nameServerIP1 /tsigAlgorithm	The hash algorithm used to generate the TSIG key. Valid values are: ▪ hmac-md5 ▪ hmac-sha1 ▪ hmac-sha224 ▪ hmac-sha256 ▪ hmac-sha384 ▪ hmac-sha512	String. Default is hmac-md5. Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList/ nameServerIP2/ip	The IP address of the first backup name server for the source zone.	IPv4 or IPv6 address. Optional for creation or update of a secondary zone.
nameServerIpList/ nameServerIP2/tsig Key	If TSIG is enabled for this name server, the name of the TSIG key.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList/ nameServerIP2 /tsigKeyValue	If TSIG is enabled for this name server, the TSIG key's value.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList/ nameServerIP2 /tsigAlgorithm	The hash algorithm used to generate the TSIG key. Valid values are: ▪ hmac-md5 ▪ hmac-sha1 ▪ hmac-sha224 ▪ hmac-sha256 ▪ hmac-sha384 ▪ hmac-sha512	String. Default is hmac-md5 Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList /nameServerIP3/ip	The IP address of the second backup name server for the source zone.	IPv4 or IPv6 address. Optional for creation or update of a secondary zone.
nameServerIpList /nameServerIP3/tsig Key	If TSIG is enabled for this name server, the name of the TSIG key.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.

Field	Description	Type
nameServerIpList /nameServerIP3 /tsigKeyValue	If TSIG is enabled for this name server, the TSIG key's value.	String. Required for creation or update of a secondary zone if TSIG is enabled for this name server.
nameServerIpList /nameServerIP3 /tsigAlgorithm	The hash algorithm used to generate the TSIG key. Valid values are: ▪ hmac-md5 ▪ hmac-sha1 ▪ hmac-sha224 ▪ hmac-sha256 ▪ hmac-sha384 ▪ hmac-sha512	String. Default is hmac-md5. Required for creation or update of a secondary zone if TSIG is enabled for this name server.

Transfer Status Details DTO

The Transfer Status Details contains the Zone Transfer Status Details.

Table 16 Transfer Status Details DTO

Field	Description	Type
lastRefresh	Displays when the last transfer attempt or refresh was.	String. Date/Time formatted in ISO 8601 format, UTC offset based on customer-specified time zone
nextRefresh	Displays when the next transfer attempt or refresh is.	String. Date/Time formatted in ISO 8601 format, UTC offset based on customer-specified time zone
lastRefreshStatus	Displays the status of the last transfer that was attempted. Valid values are: ▪ IN_PROGRESS, ▪ FAILED ▪ SUCCESSFUL	String. Date/Time formatted in ISO 8601 format, UTC offset based on customer-specified time zone
lastRefreshStatusMessage	Displays the last transfer's status message. This is currently shown as failure reason.	String.

Zone DTO

The Zone DTO is the data structure returned for the *Get Zone Metadata* call.

Table 17 Zone DTO

Field	Description	Type
properties	The basic metadata for any zone.	*Zone Properties DTO*.
restrictIpList	The list of IP ranges that are allowed to use AXFR to transfer primary zones out.	List of *Restrict IP DTO*s. Only present if this is a primary zone.
primaryNameServers	The primary name servers that are the source of a secondary zone.	*Name Server IP List DTO*. Only present if this is a secondary zone.
originalZoneName	The name of the zone that is the source of an alias zone.	Domain name. Only present if this is an alias zone.
registrarInfo	Information about the name server configuration for this zone.	*Registrar Info DTO*. Only present if this is a primary zone.
tsig	The TSIG information for the primary zone.	*TSIG DTO*. Only present if this is a primary zone.
notifyAddresses	The addresses that are notified when updates are made to the primary zone.	List of *Notify Address DTO*s. Only present if this is a primary zone.
transferStatusDetails	The zone transfer status details.	*Transfer Status Details DTO.*

Registrar Info DTO

The Registrar Info DTO holds the domain name registry information for a Primary Zone.

Table 18 Registrar Info DTO

Field	Description	Type
registrar	The name of the domain registrar.	String
whoisExpiration	The date when the domain name registration expires.	Date when the domain registration expires.
nameServers/ok	List of UltraDNS name servers that are configured for this domain.	List of domain names.
nameServers/unknown	List of name servers that are configured for this domain, but are not UltraDNS-managed name servers.	List of domain names.
nameServers/missing	List of UltraDNS name servers that should be configured for this domain, but are not.	List of domain names.
nameServers/incorrect	List of any obsolete UltraDNS name servers that are still configured for this zone.	List of domain names.

Zone List DTO

The Zone List DTO wraps the zones returned for a *List Metadata for Zones* call, and the metadata for the request.

Table 19 Zone List DTO Structure

Field	Description	Type
zones	List of the returned zones. Each item in the list matches the zone DTO described above.	List of *Zone DTOs*.
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (false) or descending (true).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of zones returned.	Integer.

Zone DNSSEC APIs

DNS Security Extensions (DNSSEC) refers to a set of security extensions to DNS, which provide DNS clients (resolvers) with origin authentication of DNS data, authenticated denial of existence, and data integrity.

The REST API allows you to sign, unsign, and re-sign a zone, as well as get current DNSSEC information and status for a zone.

On The Fly Signing

As of September 2019, a new type On the Fly signing will be available for all Neustar customers. This new signing method will apply to any newly created Zones, or if an existing zone is signed for the first time. When utilizing On the Fly signing, you are able to sign Traffic Management Pools as well as Advanced records, which have historically been restricted. Furthermore, you no longer need to Re-Sign a zone after making changes, as the zone will now automatically resign. ECDSAP256 will be used as the new algorithm as opposed to the previous RSA_SHA256.

Existing signed zones are not impacted by this change and will continue to be signed using the legacy DNSSEC signer. For zones that already have their DS record delegated, you will not able to convert to On the Fly signing at this time. We will provide a procedure for such conversion at a later date.

It is important to note, that once you sign a zone using the new On the Fly method, you are not able to revert back to using the legacy signer to sign your zones.

Sign a Zone

Signing a zone means adding DNSSEC security to the zone.

Signing a zone is a POST call (you are creating/adding a DNSSEC signature for the zone) and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/dnssec
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

- If update happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to change security on {zoneName}.
- If {zoneName} does not exist.
- If the specified zone has Sitebacker, Traffic Controller, or Directional pools (advanced services do not currently allow for signed zones).

Unsign a Zone

Unsigning a zone means removing a DNSSEC security signature from the zone.

Unsigning a zone is a DELETE call and is generated as follows:

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/dnssec
```

Parameters: None

Body: Can include the following optional field. The "Content-Type: application/json" header is required.

Field	Description	Type
changeComment	An optional field allowing users to create a comment for a zone operation using up to 512 characters of free text, which can be viewed and searched for via the Audit Log Report. Not applicable for Batch or JSON Patch calls. Additionally, the use of a colon (:) is prohibited.	String.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, a Status Code 202 is returned with a status response message of Pending along with an X-Task-Id header in body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to change security on {zoneName}.
- If {zoneName} does not exist.

Resign a Zone

When a zone is signed, modifications to the zone are not made publically available until the signatures for the zone are regenerated. This process is called Resigning a Zone. In addition to resigning after changes, some security experts recommend periodic resigning of zones, even if there are no modifications.

As of September 2019, zones that are signed by On the Fly signing method will no longer need to be manually signed, as the new signing method automatically resigns a zone when any changes are made to it.

Resigning a zone is a PUT or a PATCH call and is generated as follows:

Method and URI:

```
PUT/PATCH https://api.ultradns.com/zones/{zoneName}/dnssec
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If update happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to change security on {zoneName}.
- If {zoneName} does not exist.
- If {zoneName} is not currently signed.

Get DNSSEC Details for a Zone

The Get DNSSEC Details provides detailed information regarding the current DNSSEC status of a zone. A Time Stamp will be returned when a GET DNSSEC call is performed, and will be provided in the dnssecstatus response. The time stamp will be displayed as <last_modified> in the default XML date format (for example: “2015-11-23T10:39:59Z”), but can be reformatted via the ZoneDataeTime tool in Java 8.

The Get DNSSSEC Details call is a GET call and is generated as follows:

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/dnssec
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *DNSSEC Info Response DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to view security on {zoneName}.
- If {zoneName} does not exist.

DNSSEC Info Response DTO

The DNSSEC Info Response DTO returns the current DNSSEC details for the identified zone.

Table 20 DNSSEC Info Response DTO

Field	Description	Type
status	Current DNSSEC status of the zone. Valid values include SIGNED or UNSIGNED.	Object.
policy	The standard rules applied for the keys for this zone.	*Policy DTO.*
lastModifiedDateTime	The last date and time the zone was modified, represented in ISO8601 format.	String, Date/Time formatted in ISO 8601 format.
lastModifiedZoneDateTime	The last date and time the zone was modified.	String, date/time formatted in ISO 8601 format. Returned in GET responses for zone information.
resignNeeded	Specify if a zone resign is needed for an already signed zone. true = yes false = no	Boolean.

Table 21 Policy DTO

Field	Description	Type
policy/securityType	The security type imposed by policy. Valid values include: ▪ NONE ▪ NSEC ▪ NSEC3 ▪ NSEC3_OPT_OUT ▪ NSEC_ON_THE_FLY	Object.
policy/rrsigSignatureDuration	The period of time (in days) for which an RR Signature is valid.	Integer.
policy/nsec3Parameters	The standard parameters used for NSEC3 records.	*NSEC3 Parameters DTO.*
policy/keyPolicy	The standard policy used to define the ZSK and KSK.	Array of *Key Policy DTO*s.
keys	The current keys defined in the zone. The **dnsKeyRecord** will only be returned in a zone is signed using the On_the_Fly signing method.	Array of *Key DTO*s.

Table 22 NSEC3 Parameters DTO

Field	Description	Type
policy/nsec3Parameters/salt	Indicator that a salt value has been system generated.	String.
policy/nsec3Parameters/optOutFlag	Indicates whether this NSEC3 policy may cover unsigned delegations.	Integer.
policy/nsec3Parameters/iterations	Number of times the hash value is generated and applied.	Integer.
policy/nsec3Parameters/hashAlgorithm	The hash algorithm used in the calculation of the full hash value.	String.

Table 23 Key Policy DTO

Field	Description	Type
policy/keyPolicy/[n]/type	Key Policy Type for the zone. Valid values are KSK or ZSK.	Object.
policy/keyPolicy/[n]/bitLength	The number of bits in the specified key.	Integer.
policy/keyPolicy/[n]/keyRolloverFrequency	The Key Rollover Frequency established by the key policy for this zone.	Integer.
policy/algorithm	Crypto Algorithm Used.	String.

Table 24 Key DTO

Field	Description	Type
keys/[n]/type	Type of key. Valid values are KSK or ZSK.	Object.

Field	Description	Type
keys/[n]/status	Key Status. Valid values are: ▪ CURRENT (key is currently being used to sign the zone) ▪ EXPIRED (key was, but is no longer being used) ▪ FUTURE (key will be used to sign the zone in the future) ▪ UNKNOWN	String.
keys/[n]/publicKey	Public Key	String.
keys/[n]/nextRoll	The date on which the next key roll is scheduled to occur. Based on the keyRolloverFrequency.	Date/time in ISO 8601 format. UTC offset based on customer-specified time zone.
keys/[n]/keyRolloverFrequency	Key Rollover Frequency	Integer. Number of days.
keys/[n]/keyId	Key id	String.
keys/[n]/dsRecords	DS records	Array of String. Only applicable to KSK key.
keys/[n]/created	Creation Date	Date/time in ISO 8601 format. UTC offset based on customer-specified time zone.
keys/[n]/bitLength	Bit Length	Integer.

JSON Example – Get DNSSEC Details Response – On_the_Fly Signing


```
{
    "status": "SIGNED",
    "policy": {
        "algorithm": "ECDSAP256SHA256",
        "securityType": "NSEC_ON_THE_FLY",
        "rrsigSignatureDuration": 15552000000,
        "keyPolicy": [
            {
                "type": "ZSK",
                "bitLength": 256,
                "keyRolloverFrequency": 180
            },
            {
                "type": "KSK",
                "bitLength": 256,
                "keyRolloverFrequency": 365
            }
        ]
    },
    "keys": [
        {
            "type": "ZSK",
            "bitLength": 256,
            "keyRolloverFrequency": 180,
            "status": "CURRENT",
```

```
            "created": "2020-07-08T06:46:21Z",
            "nextRoll": "2021-01-04T06:46:21Z",
            "keyId": 45586,
            "publicKey":
"jRcwwuZGW4KrB3Tr20HHbLBmaMMFY8MFRd4ON1io2G1AaN5DyWDOjqdiPup8eZSm3CpCCTSfrx/H
Qc5inuOgfg==",
            "dnsKeyRecord": "nsec_on_the_fly.com. 100 IN DNSKEY 256 3 13
jRcwwuZGW4KrB3Tr20HHbLBmaMMFY8MFRd4ON1io2G1AaN5DyWDOjqdiPup8eZSm3CpCCTSfrx/HQ
c5inuOgfg=="
        },
        {
            "type": "KSK",
            "bitLength": 256,
            "keyRolloverFrequency": 365,
            "status": "CURRENT",
            "created": "2020-07-08T06:46:21Z",
            "nextRoll": "2021-07-08T06:46:21Z",
            "keyId": 48927,
            "publicKey":
"e5GJmifYXx1NFAVYA11jCCUTM4nXE+9FeyfUy3l3xreiS3RTAYIWRerXbIkTXAmVmokSSdihUnTH
+91R0OfWLg==",
            "dsRecords": [
                "48927 13 1 108695D72E4397854AAABE489CC1A36A64872F1A",
                "48927 13 2
97D96C00733A3FA59A1C71393FA4ED2261A6EDA78F82E7EC18AB3BE750A94F06"
            ],
            "dnsKeyRecord": "nsec_on_the_fly.com. 100 IN DNSKEY 257 3 13
e5GJmifYXx1NFAVYA11jCCUTM4nXE+9FeyfUy3l3xreiS3RTAYIWRerXbIkTXAmVmokSSdihUnTH+
91R0OfWLg=="
        }
    ],
    "lastModifiedDateTime": "2020-07-08T06:46:21Z",
    "lastModifiedZoneDateTime": "2020-07-08T06:46:21Z",
    "resignNeeded": false
}
```


Signer Error Messages

The following table displays the error messages for possible zone signing issues, along with the updated system error message that will be displayed.

Table 25 Signer Processing Error Messages

HTTP Code	Operation Type	Response Code	Current Response Message
500 – Internal Server Error	Sign Zone	1837	Could not complete sign zone request. Please retry request after some time or else contact the Ultra DNS Customer Support team.
	Unsign Zone		Could not complete unsign zone request. Please retry request after some time, or else contact the Ultra DNS Customer Support team.

HTTP Code	Operation Type	Response Code	Current Response Message
500 – Internal Server Error	Get DNSSEC	1837	Could not get dnssec status. Please retry request after some time or else contact the Ultra DNS Customer Support team.
500 – Internal Server Error	Get DomainDNSSECPolicies	1837	Could not get domain dnssec policies. Please retry request after some time or else contact the Ultra DNS Customer Support team.

Table 26 Signer Validation Error Messages

HTTP Code	Operation Code	New Response Code	New Response Message
400 – Bad Request	Sign Zone	1881	We are not able to process sign zone requests at this time. Please retry request again.
400 – Bad Request	Unsign Zone	1882	We are not able to process unsign zone requests at this time. Please try again.

Table 27 Signer User Validation Error Messages

HTTP Code	Operation Code	New Response Code	Response Message
400 – Bad Request	Sign Zone	1883	Error in signing zone. Missing SOA record for zone <zoneName>.
400 – Bad Request	Sign Zone	1884	Error in signing zone. Too many SOA records in zone <zoneName>; expected 1, found <soaCountFound>.
400 – Bad Request	Sign Zone	1885	Error in signing zone. Too many pending SOA records in zone <zoneName>; expected 1, found <soaCountFound>.
400 – Bad Request	Unsign Zone	1886	Error in unsigning zone. Missing SOA record for zone <zoneName>.
400 – Bad Request	Unsign Zone	1887	Error in unsigning zone. Too many SOA records in zone <zoneName>; expected 1, found <soaCountFound>.
400 – Bad Request	Unsign Zone	1888	Error in unsigning zone. Too many pending SOA records in zone <zoneName>; expected 1, found <soaCountFound>.

Zone Snapshot and Restore APIs

In UltraDNS, a backup is also known as a Snapshot. A zone snapshot represents the state of a zone (i.e. primarily its RRSet configuration) at the time the Snapshot is created.

Performing a zone Restore uses the most recent zone snapshot, and overwrites the zone's current configuration with that of the one stored in the Snapshot. The zone snapshot can be restored at any point in time, as long as the zone meets the required criteria.

- Snapshot and Restore only supports primary zones.
- The zone should not have more than 50,000 records, including the allowed pool's resource records.

For additional details on how to use the Zone Snapshot and Restore (ZBR), please refer to the Zone Snapshot and Restore API Guide on the support page for additional documentation.

Apex Alias

Apex Alias (APEXALIAS) provides a way for domain administrators to provide DNS CNAME functionality at the apex (or root) of a domain. Traditionally, the use of CNAMES at the apex of a domain have been discouraged and eventually forbidden by the protocol specification. However, modern usage of CDNs and outsourced cloud environments have driven the need for the ability to point the apex of a domain to resources outside of the domain. The Apex Alias addresses this need, by introducing a custom record for domain owners to maintain this linkage.

At query and resolution, the UltraDNS resolvers will chase the Apex Alias, in the same way that a recursing resolver would, and return the results of the chase in the answer to the query. Apex Alias functionality supports both IPv4 and IPv6 address resolution, returning A and AAAA records as appropriate. UltraDNS has also built in support for the edns0-client-subnet (ECS) specification, meaning that caching resolvers that pass in the client subnet will be able to take advantage of ECS functionality when querying for the zone apex.

Creating the Apex Alias

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/APEXALIAS/{zoneName}
```

Parameters: None

Body: Must include an *RRSet DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you try to sign a zone with apex alias.
- When the record type can only be created at the apex of the domain.
- HostName and pointsTo cannot be same.
- pointsTo must be a fully qualified domain name.
- This record type cannot be created on a signed zone.
- Only one APEX Alias record can be configured per zone.
- This record type cannot be created when both A and AAAA records exist at zone apex.
- A/AAAA cannot be added so that there are apex alias and both A and AAAA at apex.

Reading the Apex Alias

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/APEXALIAS/{zoneName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Data is not found.
- Insufficient permissions: User cannot access object.
- Zone does not exist in the system.

JSON Example: Reading the Apex Alias

```
{
   "zoneName": "ultratest.biz”
   "rrsets": [
      {
         "ownerName": "ultratest.biz"
         "rrtype": "APEXALIAS (65282)"
         "ttl": 300,
         "rdata": [
            "mywebfront.mysecretcdn.com"
            ]
         }
      ]
      "queryInfo": (
         "sort": "OWNER",
         "reverse": false,
         "limit": 100
         }
      "resultInfo": {
         "totalCount": 1,
         "offset": 0,
         "returnedCount": 1
      }
}
```

SSHFP Records

A Secure Shell Fingerprint (SSFHP) record is a resource record that identifies the SSH keys (which provide secure remote log and network services over an insecure network) that are associated with a host or domain name. The SSHFP record can be used when a public key is not recognized, and if accepted, will be saved locally and used for verification for subsequent connections.

Per RFC-4255, "Upon connection to an SSH server, the SSH client MAY look up the SSHFP resource record(s) for the host it is connecting to. If the algorithm and fingerprint of the key received from the SSH server match the algorithm and fingerprint of one of the SSHFP resource record(s) returned from DNS, the client MAY accept the identity of the server."

The SSHFP record consists of an alrogirthm number, a fingerprint type, and the fingerprint of the public host key.

Create SSHFP Record

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/SSHFP/{zoneName}
```

Parameters: None

Body: Must include an *RRSet DTO*, and the following SSHFP DTO:

Field	Description	Type
Algorithm	The first integer displayed. Describes the type of Algorithm to be used.	Valid values are: ▪ **1** – RSA; ▪ **2** – DSS; ▪ **3** – ECDSA; ▪ **4** – Ed25519
Type	The Algorithm type used to hash the public key.	▪ **1** – SHA-1; ▪ **2** – SHA-256
Fingerprint	The hexadecimal representation of the hash result, as text.	Valid hexadecimal string.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or if it's not a {zoneName} you have access to.
- If an invalid Input is provided.
- If you don't have permission to create SSHFP Records.
- If a resource record or {zoneName} of the same type already exists

JSON Example: Create SSHFP Records

```
{
    "rdata": [
        "1 1 6E657573746172"
    ]
}
```

Get SSHFP Record

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/SSHFP
```

OR

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/SSHFP/{ownerName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Data is not found.
- Insufficient permissions: User cannot access object.
- Zone does not exist in the system.

JSON Example: Get SSHFP Records

```
{
   "zoneName": "ultratest.biz”
   "rrsets": [
      {
         "ownerName": " www.sshfp.com.ultratest.biz"
         "rrtype": "SSHFP (44)",
            "ttl": 100,
            "rdata": [
                "1 2 6E657573746172"
            ]
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
```

```
}
```

Delete SSHFP Record

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/SSHFP/{zoneName}
```

Parameters: None

Body: None

Response: If delete happens immediately, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- Data is not found.
- Insufficient permissions: User cannot access object.
- Zone does not exist in the system.

DS Records

The Delegation Signer (DS) record refers to a DNSKEY resource record, and is used in the DNSKEY authentication process. The DS resource record is inserted at a zone cut (i.e., a delegation point) to indicate that the delegated zone is digitally signed and that the delegated zone recognizes the indicated key as a valid zone key for the delegated zone. By authenticating the DS record, a resolver can then authenticate the DNSKEY resource record, to which the DS record points to.

The RDATA for a DS record consists of a two-octet **Key Tag** field, a one-octet **Algorithm** field, a one-octet **Digest Type** field, and a **Digest** field.

For example, the DS resource record for "example.com" is stored in the "com" zone (the parent zone), rather than in the "example.com" zone (the child zone). The corresponding DNSKEY resource record is stored in the "example.com" zone (the child zone). This simplifies DNS zone management and zone signing. For additional information, please refer to RFC-3658.

Create DS Record

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/DS/{zoneName}
```

Parameters: None

Body: Must include an *RRSet DTO*, and the following DS DTO:

Field	Description	Type
Key Tag	The Key Tag field contains the key tag value of the DNSKEY RR that validates this signature, in network byte order.	Integer.
Algorithm	The Algorithm type used to hash the public key. For the full list of Algorithm types, please refer to	▪ **8** – RSA/SHA -256 ▪ **13** – ECDSA Curve P-256 with SHA-256.
Digest Type	The digest refers to the DNSKEY resource record. The Digest Type Identifies the algorithm used to construct the digest.	▪ **1 –** SHA-1 ▪ **2** – SHA-256
Digest	The digest is calculated by concatenating the canonical form of the fully qualified owner name of the DNSKEY RR with the DNSKEY RDATA, and then applying the digest algorithm.	Hexadecimal string – 40 characters in length. Not returned on GET call.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or if it's not a {zoneName} you have access to.

- If an invalid Input is provided.
- If you don't have permission to create DS Records.
- If a resource record or {zoneName} of the same type already exists

JSON Example: Create DS Record

```
{
   "ttl":500,
   "rdata":["1000 8 1 A94A8FE5CCB19BA61C4C0873D391E987982FBBD3"]
}
```

Get DS Record

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/DS
```

OR

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/DS/{ownerName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Data is not found.
- Insufficient permissions: User cannot access object.
- Zone does not exist in the system.

JSON Example: Get DS Records

```
{
   "zoneName": "ultratest.biz”
   "rrsets": [
      {
         "ownerName": " www.ultradstest.com.ultratest.biz"
         "rrtype": "DS (43)",
            "ttl": 800,
            "rdata": [
                "1000 8 1 A94A8FE5CCB19BA61C4C0873D391E987982FBBD3"
            ]
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
```

```
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```

Delete DS Record

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/DSs/{zoneName}
```

Parameters: None

Body: None

Response: If delete happens immediately, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- Data is not found.
- Insufficient permissions: User cannot access object.
- Zone does not exist in the system.

Resource Record Sets

A zone contains one or more Resource Records (RRs). A grouping of Resource Records of the same type at the same domain name (for example, all A Records at example.com.) is called a Resource Record Set (*RRSet)*. The domain name containing the RRSet is the *owner* of the RRSet; the domain name is the *ownerName*.

Resource Record Set (RRSet) DTOs

This section describes the RRSet DTOs used for the RRSet calls defined below.

RRSet DTO

The basic unit for resource record manipulation in the REST API is the RRSet. An RRSet contains the data for all resource records present at the same owner name (label), and with the same type and class (all records in UltraDNS have IN class).

- The UltraDNS REST API enforces the same TTL for all records in an RRSet (with the exception of a directional pool).
- Rather than trying to specify a custom structure for each different resource record type, the data for all resource records is represented by an "rdata" field. This field's contents map to the data supplied in the BIND presentation format for a resource record.
- If an RRSet is associated with an UltraDNS pool, the “profile” field is present and contains pool specific attributes.

The RRSet DTO is used in the List (GET), Create (POST), and Update (PATCH) RRSets API calls. When a user is submitting an RRSet DTO to the server, the owner name and rrtype are not required, because they are specified by the URI. If present, they will be ignored.

Table 28 RRSet DTO

Attribute	Description	Type	Required?
ownerName	The domain name of the owner of the RRSet. Can be either: ▪ Fully Qualified Domain Name (FQDN) ▪ Relative Domain Name. If a FQDN, it must be contained within the zone name FQDN.	String.	NO for PUT, POST, or PATCH calls. Ignored if present. Not present (and ignored if present) if this RRSet is embedded inside of an Owner structure.
rrtype	Resource Record Type for the RRSet. Must be formatted as the well-known resource record type (A, AAAA, TXT, etc.) and the corresponding number for the type, between 1 and 65535 or a known resource record name (A, AAAA, SRV, etc.).	String.	NO for PUT, POST, or PATCH calls. Ignored if present. Not present (and ignored if present) if this RRSet is embedded inside of an Owner structure.

Attribute	Description	Type	Required?
ttl	The time to live (in seconds) for all records in the RRSet. Must be a value between 0 and 2147483647, inclusive.	Integer.	Should be included in PUT, POST or PATCH calls. If TTL is not specified, the value set at the account level (either for record type or global setting) will be used.
rdata	The data for the records in the RRSet. Must use the BIND presentation format for the specified rrtype. **For MX, NS, CNAME, PTR, and APEXALIAS record types, the rdata value cannot be relative to the zone name. It must be a FQDN.** *If rrtype is MX, entering "0 ." will create a **NULL** MX record, implying that "No Service" is available for the server / record.*	Array.	REQUIRED for PUT, POST, or PATCH calls. Must include BIND formatted data. *If a null MX record exists for a zone, no additional MX records are allowed. Additionally, if an MX record exists within a zone, a Null MX record cannot be created.*
systemGenerated	Indicates whether the record in an rdata list is system generated or not. It represents the Boolean value that corresponds to the record in the list (in same order the records are returned).	Array (Boolean).	Will be returned for GET requests when used as the query parameter **systemGeneratedStatus**. If the query parameter is set to true, then the attribute name will be returned in the response. Ignored for PUT/PATCH/POST requests.

JSON Example: Returning an A Record in a Zone

```
{
    "ownerName": "a.domain.name.",
    "rrtype": "A (1)",
    "ttl": 300,
    "rdata": [
        "1.2.3.4"
    ]
}
```

JSON Example: Create a NULL MX Record

```
{
    "ownerName": "a.domain.name.",
    "rrtype": "MX (15)",
    "ttl": 300,
    "rdata": ["0 ."]
}
```

JSON Example: Return an A Record in a Zone with systemGeneratedStatus true

```
GET:
https://api.ultradns.com/zones/{zoneName}/rrsets/A/a.domain.name.?systemGener
atedStatus=true

{
    "ownerName": "a.domain.name.",
    "rrtype": "A (1)",
    "ttl": 300,
    "rdata": [
        "1.2.3.4"
    ],
    "systemGenerated": [false]
}
```

RRSetList DTO

This is returned when retrieving multiple RRSets from the server. It is not used for creating or updating RRSets.

Table 29 RRSetList DTO

Field	Description	Type
zoneName	The FQDN for the zone.	String.
rrsets	The list of *RRSet DTOs*.	Array.
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (false) or descending (true).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

List all RRSets in a Zone

This call provides a list of all RRSets in a zone, or if specific query parameters are used, a list of the RRSets in a zone that match the provided criteria.

The {zoneName} identified in the call should be the name of the domain whose RR sets you want to return.

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets
```

Parameters: The following parameters are available to be sent with the call.

Table 30 List RRsets in Zone

Parameter	Description	Type
q	The query used to construct the list. Query operators are: ▪ **TTL**: Exact match for the TTL value. Only valid for **RECORDS** RRSets type; ignored for the other kinds of RRSets (see below). ▪ **Owner**: Partial match to an owner name. Valid for all RRSets. ▪ **Value**: Partial match for the rdata for a resource record. Only valid for the **RECORDS** RRSet type; ignored for other kinds of RRSets (see below). ▪ **Kind** (defaults to ALL): The kind of RRSets or Pools that will be returned. One or more types can be specified if separated by commas. Valid values are: ○ ALL (returns all kinds) ○ RECORDS ○ POOLS (returns all pool kinds) ○ RD_POOLS ○ DIR_POOLS ○ SB_POOLS ○ TC_POOLS	String.
offset	The position in the list for the first returned element (0 based). Default is 0.	Integer.
limit	The maximum number of rows requested. Default is 100.	Integer.
sort	The sort column used to define the order of the list. Valid values are: ▪ OWNER (default) ▪ TTL ▪ TYPE (specifies Record type).	String.
reverse	List is sorted in *Ascending* order with **false** by default. **true** will sort the list in *Descending* order by the sort column specified (or by OWNER if no sort value is entered).	Boolean. Default is false..
systemGeneratedStatus	Used to indicate whether the records in rrdata are system generated or not. Returns the **systemGenerated** attribute in an RRSet response.	Boolean Default is false.

JSON Example: List all RRSets for a Zone

```
{
  "zoneName": "primary-example.com.",
  "rrSets": [
    {
      "ownerName": "arecord.primary-example.com.",
      "rrtype": "A (1)",
      "ttl": 500,
      "rdata": [
        "1.1.1.1"
      ]
```

```
    },
    {
      "ownerName": "arecord.primary-example.com1.primary-example.com.",
      "rrtype": "A (1)",
      "ttl": 500,
      "rdata": [
        "1.1.1.1"
      ]
    },
    {
      "ownerName": "arecord.primarytest.com.primary-example.com.",
      "rrtype": "A (1)",
      "ttl": 500,
      "rdata": [
        "2.2.2.2"
      ]
    },
    {
      "ownerName": "arecord.primarytest.com.primary-example.com.",
      "rrtype": "AAAA (28)",
      "ttl": 500,
      "rdata": [
        "fdda:5cc1:23:4:0:0:0:1f"
      ]
},
```

URL Example: List RRSets in a Zone with Parameters

```
https://api.ultradns.com/zones/example.com/rrsets?q=ttl:300+owner:test+kind:R
ECORDS&sort=TYPE&reverse=true@systemGeneratedStatus=true
```

The example URL shown above does the following:

- Returns all standard RRSets (no pools) for the domain “example.com” that have a TTL of 300,
- An owner name that includes the string “test”,
- Sorts the list in descending order by record type.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RRSetList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not a valid zone name.
- If you don't have permission to read the zone.

JSON Example: List RRSets with Query Parameters

```
{
  "zoneName": "primary-example.com.",
  "rrSets": [
    {
      "ownerName": "srvrecord.primarytest.com.primary-example.com.",
```

```
      "rrtype": "SRV (33)",
      "ttl": 500,
      "rdata": [
        "1 5 81 test.primarytest.com."
      ],
      "systemGenerated": [false, false]
    },
    {
      "ownerName": "spfrecord.primarytest.com.primary-example.com.",
      "rrtype": "SPF (99)",
      "ttl": 500,
      "rdata": [
        "\"spfrecorddata\""
      ],
      "systemGenerated": [false]

    },
    {
      "ownerName": "rprecord.primarytest.com.primary-example.com.",
      "rrtype": "RP (17)",
      "ttl": 500,
      "rdata": [
        "mail1.primary-example.com. text1.primary-example.com."
      ],
      "systemGenerated": [false, true]
    },
    {
      "ownerName": "ptrrecord.primarytest.com.primary-example.com.",
      "rrtype": "PTR (12)",
      "ttl": 500,
      "rdata": [
        "test1.primarytest.com.primary-example.com."
      ],
      "systemGenerated": [false]
},
```


List all RRSets by Type

This call allows you to list all RRSets of a particular Record type for a specified zone. This differs from the above *List all RRSets in a Zone* because the above call returns RRSets of records based on the kind of set or pool, while this call returns RRSets based on the record type.

The Type can be a DNS record name (for example A, TXT, AAAA, SRV), or a number value that corresponds to a resource record type.

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{recordType}
OR
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{recordTypeNumber}
```

URL Example: List All RRSets of Type A in a Zone

```
https://api.ultradns.com/zones/example.com/rrsets/A
```

This returns all A records in the zone example.com.

URL Example: List All RRSets of Type 16 (TXT) in a Zone

```
https://api.ultradns.com/zones/example.com/rrsets/16
```

This returns all TXT (resource record type code 16) in the zone example.com.

Parameters: *Table 30 List RRsets in Zone* in zone lists the parameters available to be sent with the call.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RRSetList DTO* in the body content, containing only RRSets of the specified type.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read zones.
- If type is invalid.

List all RRSets of a Type for an Owner

This call returns a list of RRSets of a specified type for the Owner Name provided in the call.

The identified **Type** can be any of the following:

- A DNS record-type name (A, TXT, AAAA, SRV),
- A number corresponding to a resource record type (from 1 to 65535).

OR

- The special reserved word "ANY", which will return all RRSets for the specified owner name.

Owner name can be a relative name (does not include the zone name), or an absolute name (includes the zone name).

Note the following about owner names and constructing this call:

- Using a **q** query parameter with an owner name is automatically a string-match search; it will return all owner names that contain the specified string.
- When the owner name is specified as part of the URI, the search will return only the RRSets for the owner name that exactly matches the specified name.

See the examples listed below for more information.

The list all RRSets of a type for an owner is a GET call and is constructed as follows:

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

Example List RRSets of a Type for an Owner: Methods and Results

For the following example calls, assume there are A-type records defined at *test1.domain.name*, *test2.domain.name*, and *test.domain.name*. Different call construction will return different results.

URL Example 1: Return all A-type RRSets across *test*.domain.com

For the identified example scenario, the following call returns A-type Resource Record Sets for all three domain.name zones where the owner name contains the string “test”:

```
https://api.ultradns.com/zones/domain.name/rrsets/A?q=owner:test
```

URL Example 2: Return all A-type RRSets for “test.domain.com” only

If you construct the call using the same “test” string as the owner name, but in the URL instead of as a query parameter, the call looks as follows and returns only the resource record set for *test.domain.com*.

```
https://api.ultradns.com/zones/domain.name/rrsets/A/test
```

For getting the **systemGenerated** attribute in an RRSet response, the following query parameter needs to be provided:

```
https://api.ultradns.com/zones/domain.name/rrsets/A/test?systemGeneratedStatus=true
```

URL Example 3: Return all RRSets for “test.domain.com” only

Using the “ANY” term for {type} in the call returns all RRSets for the specified zone. The following two examples will return the same results. The first contains an absolute owner name, including the domain name; the second contains a relative owner name, without the domain name.

```
https://api.ultradns.com/zones/domain.name./rrsets/ANY/test.domain.name

https://api.ultradns.com/zones/domain.name/rrsets/ANY/test
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RRSetList DTO* in the body content.

JSON Example: Returned records list for ownerName *test.domain.name*

```
{
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 2,
        "offset": 0,
```

```
        "returnedCount": 2
    }
    "zoneName": "domain.name.",
    "rrsets": [
      {
        "ownerName": "test.domain.name.",
        "type": "A (1)",
        "ttl": 300,
        "rdata": [
          "9.8.7.6"
        ],
      },
      {
        "ownerName": "test.domain.name.",
        "type": "TXT (16)",
        "ttl": 300,
        "rdata": [
          "The quick brown fox jumped over the lazy dog",
          "Here is another TXT record in the RRSet"
        ],
      }
    ],
}
```


Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read zones.
- If type is invalid.
- If {ownerName} does not exist.

Create RRSet for an Owner

The Create RRSet for an Owner call requires you to send an *RRSet DTO* with the call. However, the ownerName and rrtype fields are not required because they are specified in the URI. If the DTO you send does include them, they will be ignored.

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

URL Example: Create an A-type RRSet for owner *test.domain.name*

```
https://api.ultradns.com/zones/domain.name./rrsets/A/test
```

The above example creates an A-type resource record set for the owner *test.domain.name.*

Parameters: None

Body: Must include an *RRSet DTO*.

JSON Example: RRSet DTO for an A-type RRSet for *test.domain.name*

```
{
    "ttl": 300,
    "rdata": [
        "1.2.3.4"
    ]
}
```

Response: If creation happens immediately, Status Code 201 is returned with an appropriate status message in the response body.

- If creation happens in the background, a Status Code 202 is returned with a status message of "Pending", along with an X-Task-Id header in body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to write records.
- If {type} is invalid.
- If {ownerName} is not a valid name.

Create Zone via BIND File Upload

This call creates a Zone, rather than individual records within a Zone. The Zone details must be attached in a BIND file, and attached in the body of the call when sent.

New Update – The BIND Upload of zone feature now supports the usage of whitespaces as a hostname in the BIND file. Referring to RFC-1035, newly supported behavior is described as follows – "*If an entry for an RR begins with a blank, then the RR is assumed to be owned by the last stated owner. If an RR entry begins with a <domain-name>, then the owner name is reset.*"

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets
```

Parameters: *RRSet DTO*

Body: You will need to upload/provide the partial BIND file location by changing the **KEY** parameter to **File**, and then the **Content-Type** as **multipart/form-data**. Click on the **Select Files** button to upload your BIND file.

Response: Task should happen in the background, returning a Status Code 202 along with an X-Task-ID header and status message of "Pending" in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to update {zoneName}.

Do not add any JSON content or BIND file content as text in the request body. Below is a screenshot example of how to setup the request to upload a partial BIND file.

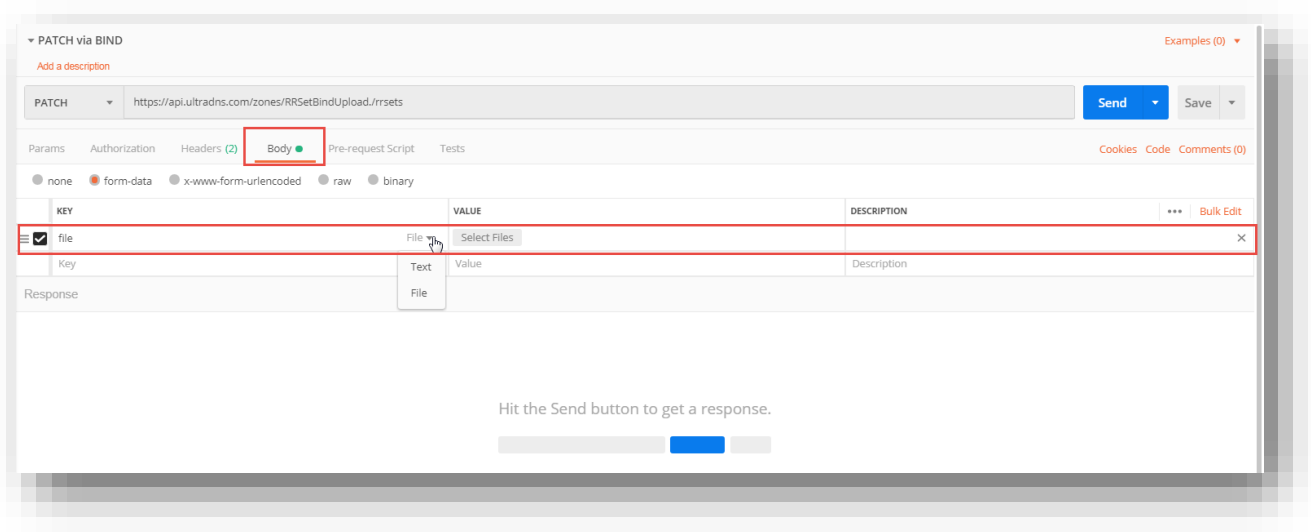

Figure 5 RRSet Patch via Bind

BIND File Example: Create Zone

```
;File created: 12/04/2019 12:07
;Record count: 20
$ORIGIN upload-test.com.
@  86400 IN SOA udns1.ultradns.net. BindZone\.testzone.neustar.biz. (
         2019041222  ;Serial
         86400    ;Refresh
         86400    ;Retry
         86400    ;Expire
         86400    ;Minimum
       )
owner3   300   IN NS ns1.nameserver.com.
@  86400 IN NS udns1.ultradns.net.
@  86400 IN NS udns2.ultradns.net.
owner1   100   IN A  12.12.12.12
owner4   400   IN CNAME cname1.com.
owner13  800   IN MB mail2.upload-test.com.
owner14  900   IN WKS   12.12.14.12 6 26
owner8   500   IN PTR   test.addr2.com.
owner7   403   IN HINFO "32bits" "windows"
owner6   402   IN MX 10 mail2.upload-test.com.
owner5   401   IN TXT   "abcdefghijklmnopqrstabcdefghijklmnopqrst"
owner12  700   IN RP mail1.addr.arpa. text1.addr.arpa.
owner15  1000  IN KEY   33 44 44 test
owner2   200   IN AAAA  2001:db8:85a3:0:0:8a2e:370:7334
owner16  1100  IN NXT   test1.upload-test.com. SSHFP
owner11  602   IN SRV   1 5 81 test.addr.com.
owner9   600   IN NAPTR 100 6 "u" "E2U+smtp"
"!^.*$!mailto:information@addr.com!i" .
```

```
owner17  1200  IN SSHFP 32 44 AD
owner10  601   IN SPF
   "aaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaa
x2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthh
yuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaax2rgthhyuiyaaa
x2rgthhyuiyaaax2rgthhyuiyaaa" "x2rgthhyuiy"
```

JSON Example: Create Zone Upload File with Blankspace

```
owner3 300   IN NS ns1.nameserver.com.
       86400 IN NS udns1.ultradns.net.
@      86400 IN NS udns2.ultradns.net.
```

The behavior of the REST API for the above example is as follows:

- For first entry, a NS record with hostname- owner3 will be created.
- For the second entry, the hostname has been left blank, so the hostname will be taken from the preceding entry, so a record with owner3 as the hostname is created. \
    - If there is no preceding entry, an error will occur.
- For the third entry, the hostname is provided as " @ ". This record will be created with the hostname that is the same as the zone name.

Create Multiple RRSets in a Zone via BIND File Upload

This call adds new RRSets in an identified zoneName with the information in an attached BIND file. Any conflicts with existing resource record sets will result in a 202 response, but no changes will be made.

The BIND file must be included in the body sent with the call.

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets
```

URL Example: Add/Updated RRSets to an Existing Zone via BIND File

```
PATCH https://api.ultradns.com/zones/domain.name./rrsets
```

Parameters: None

Body: You will need to upload/provide the partial BIND file location by changing the **KEY** parameter to **File**, and then the **Content-Type** as **multipart/form-data**. Click on the **Select Files** button to upload your BIND file.

Do not add any JSON content or BIND file content as text in the request body. Below is a screenshot example of how to setup the request to uploa a partial BIND file.

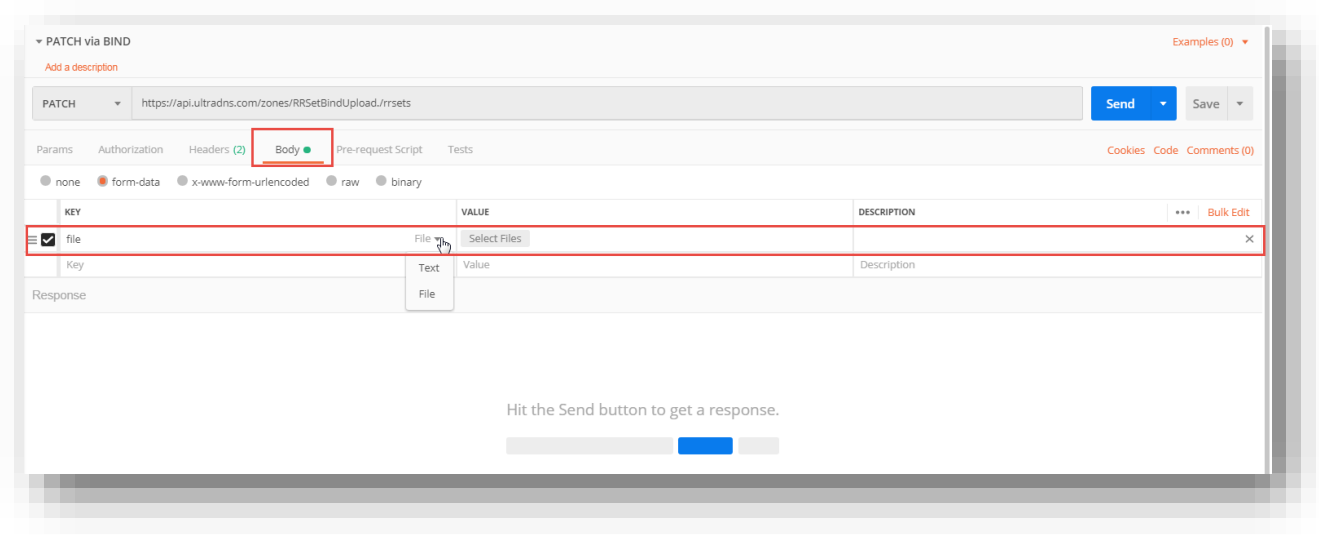

Figure 6 RRSet Patch via Bind

Response: Task should happen in the background, returning a Status Code 202 along with an X-Task-ID header and status message of "Pending" in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} does not exist.
- If you do not have permission to update {zoneName}.

Update an RRSet

This call allows you to update a set of resource records of a particular type (an RRSet) for a specified domain owner.

Be sure to specify the TTL and ALL of the record information. Any resource records not included will be removed from the RRSet, and the TTL value specified at the account level for the record type (or global TTL value) will be used.

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to update records.
- If type is invalid.
- If {ownerName} does not exist.

Partially Update an RRSet

This call allows you to update some of the information in a set of resource records of a particular type (an RRSet) for a specified domain owner.

This differs from the Update RRSet call in two ways:

1. If you do not specify the TTL, the existing TTL will continue to be used.
2. Any resource records specified will be *added* to the RRSet. Existing records are not modified or removed (except for A and CNAME records; see below note.).

There is a special case for A and CNAME resource records. Since only a single resource record is allowed in an A or CNAME RRSet, if you perform a partial update, you will modify the single resource record.

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

Parameters: None

Body: For standard XML or JSON calls, you must include an *RRSet DTO*.

For *JSON PATCH Examples*, the body must include a *JSON PATCH DTO*.

Patchable Objects for Resource Records:

- biz.neustar.ultra.rest.dto.RRSet
- biz.neustar.ultra.rest.dto.RRSet.List<String>

An Rdata element value can be patched if the index has identified each value delimited by a space in the string. Refer to the below example.

JSON Example: Update only the mail server name in the 1st MX record, or replace all values in 2nd MX record for rdata ["10 mail1.json-patch-rr-set-pr-zone.com.", "20 mail1.json-patch-rr-set-pr-zone.com."]

```
[
   {
      "op": "replace",
      "path": "/rdata/0/1",
      "value": "new.mail.server.biz."
   },
```

```
   {
      "op" : "replace",
      "path": "/rdata/1"
      "value": "30 new3.mail.server.biz"
   }
]
```

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don't have permission to update records.
- If {type} is invalid.
- If patch operation is not allowed for the given JSON pointer value.

Delete All RRSets for an Owner and Type

This call allows you to delete all resource records of a particular type (an RRSet) for a specified domain owner.

The Delete All RRSets call does not allow for the use of ANY as the {type} entry.

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

Parameters: None

Body: None

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, a Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to delete records.
- If type is invalid.
- If {ownerName} does not exist.

TTL Records Consistency in Resource Records Set

Per RFC 2181, the *Time to Live* (TTL) of all Resource Records in an RRSet must be the same value. For RRSets that (can) have multiple records and the same owner and type, the TTL behavior will act as follows.

Updating the TTL of Records via PATCH Request

Whenever a TTL value of an existing record in an RRSet is modified using a Partial Update (PATCH) request, the other records will also be updated to the newly modified TTL value. This will also create an Audit entry for each record that is being modified.

For example, the following three records are of owner “txtrecord” and type “TXT”, with rdata details and TTL values.

1. txtrecorddata1 with TTL 300
2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata3 with TTL 400*

Now, when you update the record with *txtrecorddata1* to TTL value of 600, the other two rdata records will also update their TTL values to the new 600 value.

Updating the TTL of Records via PATCH BIND File

If you are performing an update via a BIND file that contains multiple records for an RRSet (same owner and type) with different TTL values for each record, then each record will be created with the lowest TTL value amongst those RRset records in the BIND file.

For example, the following three records are of an owner “txtrecord” and type “TXT”, with rdata details and varying TTL values.

1. txtrecorddata1 with TTL value 300
2. *txtrecorddata2 with TTL value 500*
3. *txtrecorddata3 with TTL value 400*

Once the update is completed, each RRSet record will have a TTL value of 300.

Creating a New Record via Bind Upload

When you are creating a new record for an Owner and Type via a Bind upload, if the TTL value is not provided in the request, and there are pre-existing records for this specific Owner and Type, the REST API will create a new record with a TTL value that matches the lowest common TLL value amongst the existing records (of the same Owner and Type).

If there are no pre-existing records for the Owner and Type, and a TTL value is not provided at the time of the Bind upload, then the TTL value will be set to the default value.

For example, the following three existing records are of Owner “txtrecord” and Type TXT, with rdata details and TTL values.

1. txtrecorddata1 with TTL 300
2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata with TTL 400*

Now, when you create the new record for the same Owner and Type via Bind upload, the new records will have TTL value(s) of 300, and any of the pre-existing records for the same Owner and Type will have the TTL values updated to 300.

When you are creating a new record for an Owner and Type via a Bind upload, if the TTL value is provided in the request, and there are pre-existing records for this specific Owner and Type, the REST API will create new records with the provided TTL value, and will also update the TTL value of the pre-existing records (of the same Owner and Type) with the new TTL value being provided.

For example, the following three pre-existing records are of Owner “txtrecord” and Type TXT, with rdata details and TTL values:

1. txtrecorddata with TTL 300
2. *txtrecorddata with TTL 500*
3. *txtrecorddata with TTL 400*

Once the upload is complete, the three RRSet records above will be created with a TTL value of 300, and all the of the pre-existing records for the Owner and Type will also be updated to the TTL value of 300.

Updating the TTL of Records via a POST Request

When you are creating a new record for an Owner and Type, if the TTL value is not provided in the request, and there are no pre-existing records for the specific Owner and Type, then the new record will be created with the default TTL value (86400).

For example, if you add a record of Owner “txtrecord” and Type TXT but without a TTL value, the record will be assigned the TTL value of 86400.

When you are creating a new record for an Owner and Type, if the TTL is provided in the request and there are pre-existing records for the specified Owner and Type, the REST API will create a new record with the provided TTL value, and will also update the TTL value of the pre-existing records to the new TTL value.

For example, the following three pre-existing records are of Owner “txtrecord” and Type TXT, with rdata details and TTL values:

1. txtrecorddata1 with TTL 300

2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata3 with TTL 400*

If you add the new record with the TTL value of 700 via the POST request, all of the above mentioned pre-existing records will update their TTL value to 700 as well.

When you are creating a new record for an Owner and Type, if the TTL value is not provided in the request, and there are pre-existing records for the specified Owner and Type, the REST API will create a new record with the lowest TTL value amongst the pre-existing records of the specified Owner and Type.

For example, the following three pre-existing records are of Owner “txtrecord” and Type TXT, with rdata details and TTL values:

1. txtrecorddata1 with TTL 300
2. *txtrecorddata2 with TTL 500*
3. *txtrecorddata3 with TTL 400*

If you add a new record without providing the TTL value via a POST request, the new record will be added with the lowest TTL value from the pre-existing records, so in this scenario, the TTL value 300.

Resource Distribution Pools

Resource Distribution (RD) Pools are used to define rules for returning multiple A or AAAA records for a given owner name. There are three different ordering rules possible:

- **Fixed** (records appear in the same order all the time).
- **Random** (order of the records is random on each request).
- **Round robin** (the order of the records changes on each request, in order).

Profile

All pools are implemented as information added to *Resource Record Sets* (RRSets). This additional information is specified in a section within the label profile. Every profile contains an entry with the label @context. The value of @context is a URI that uniquely identifies the type of the pool.

The URI for an RD pool is `http://schemas.ultradns.com/RDPool.jsonschema`.

The other fields in the profile for an RD pool are:

Table 31 RD Pool Profile Fields

Field	Description	Type
order	The order the records will be returned in.	From one of: FIXED, RANDOM, ROUND_ROBIN
description	An optional description of the RD pool.	String, less than 255 characters. If not specified, the owner name for the RRSet will be used.

JSON Example: Resource Distribution Pool

```
{
    "zoneName": "andria.com",
    "rrsets": [
        {
            "ownerName": "redredrobin.andria.com.",
            "rrtype": "A (1)",
            "ttl": 86400,
            "rdata": [
                "198.16.1.22",
                "192.168.2.56"
            ]
         }
   ],
            "profile": {
                "@context": "http://schemas.ultradns.com/RDPool.jsonschema",
                "order": "ROUND_ROBIN",
                "description": "T. migratorius"
            },
    "queryInfo": {
        "q": "kind:POOLS",
        "sort": "OWNER",
```

```
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```


RD Pools can only be defined for RRSets of type A (1) or AAAA (28). It is an error to define an RD Pool for other RRSet types.

Multiple A or AAAA records for a single owner can only be defined if the owner is a pool. It is an error to define multiple RData for an owner of type A or AAAA if the owner is not a pool.

It is legal to define an RD Pool with zero or one Rdata records.

The order of the records in the pool is determined by the order in which they are listed in the rdata array Listing RD Pools. See *Figure 7 Listing RD Pools*.

Listing RD Pools

RD Pools are listed just like standard resource record sets. As described in *Table 30 List RRsets in Zone*, use the query operation **kind** for the **q** query parameter to control which resource record set types are returned.

Table 32 RD Pool: Kind values

Value	Meaning
ALL	All pools and records (same as RECORDS,POOLS)
RECORDS	Only resource Records.
POOLS	All Pools.
RD_POOLS	Only RD Pools.

- These values can be comma-separated if you wish to specify more than one.
- If kind is not specified, it defaults to the value ALL.

URL Example: Return all RD Pools in zone andria.com.

```
https://api.ultradns.com/zones/andria.com/rrsets?q=kind:RD_POOLS
```

This will return all RD Pools in the zone *andria.com*.

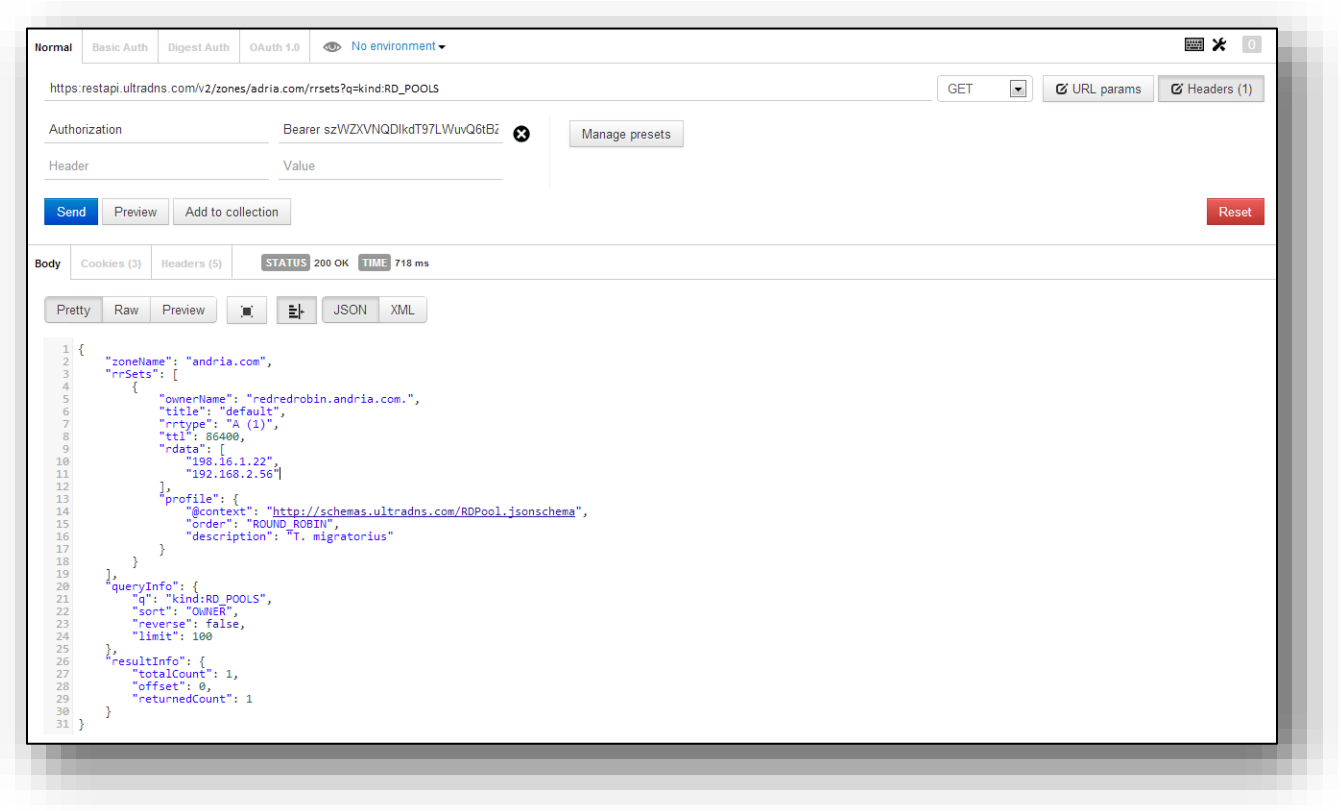

Figure 7 Listing RD Pools

Displaying RD Pools

When an owner that represents an RD Pools is returned, the profile information must be included.

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: None.

Response: If task completes, Status Code 200 OK is returned with *RD Pool Profile Fields* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.

JSON Example: GET RD Pool


```
{
     "zoneName": "primary-example.com.",
     "rrSets": [
       {
         "ownerName": "rdpool.primarytest.com.primary-example.com.",
         "rrtype": "A (1)",
         "ttl": 300,
         "rdata": [
           "1.2.3.4",
           "2.4.6.8",
           "9.8.7.6"
         ],
         "profile": {
           "@context": "http://schemas.ultradns.com/RDPool.jsonschema",
           "order": "RANDOM",
           "description": "This is a great RD Pool"
         }
       }
     ],
     "queryInfo": {
       "sort": "OWNER",
       "reverse": false,
       "limit": 100
     },
     "resultInfo": {
       "totalCount": 1,
       "offset": 0,
       "returnedCount": 1
     }
}
```


Create RD Pools

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with RD pool profile info, or a *JSON PATCH DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.


```
{
    "ttl": 300,
    "rdata":
    [
       "1.2.3.4",
```

```
        "2.4.6.8",
        "9.8.7.6"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/RDPool.jsonschema",
        "order": "RANDOM",
        "description": "This is a great RD Pool"
    }
}
```

Update RD Pools

For partial updates (PATCH) that do not affect the order or description, the profile section is not required.

For full updates, (PUT) the RD Pool profile must be fully specified.

To change the order of records in the RD Pool, or to remove a record from the RD Pool, a full update (PUT) must be performed.

To add a record to an existing RD Pool, a partial update (PATCH) can be performed.

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with RD pool profile info, or a *JSON PATCH DTO*.

Response: If the task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If the task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.

JSON Example: Update RD Pool

```
{
    "ttl": 86400,
    "rdata": [
              "206.204.52.32",
              "216.12.145.20",
              "1.2.3.6",
              "1.2.3.5"
             ],
    "profile": {
            "@context": "http://schemas.ultradns.com/RDPool.jsonschema",
            "order": "ROUND_ROBIN",
            "description": "www"
```

```
                }
}
```

Partially Update RD Pools

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with RD pool profile info, or a *JSON PATCH DTO*.

Patchable Objects for RD Pools:

- biz.neustar.ultra.rest.dto.RDPoolProfile

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don't have permission to perform a Partial Update of RD Pools.

JSON Example: Partial Update an RD Pool

```
{
      "ttl": 86400,
      "rdata": [
                "10.10.10.1"
               ],
      "profile": {
                "@context": "http://schemas.ultradns.com/RDPool.jsonschema",
                "order": "ROUND_ROBIN",
                "description": "www"
               }
}
```

Delete an RD Pool

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: None

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, a Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to delete records.
- If the {rrType} is invalid.
- If {ownerName} does not exist.

Converting To and From RD Pools

Existing A or AAAA Records to RD Pools

To convert an existing owner of a single A or AAAA record to an RD pool, perform an update (PUT or PATCH) and include the profile information for an RD Pool.

Existing RD Pool to an A or AAAA Record

To convert an RD Pool to an owner of a single A or AAAA record, perform a full update (PUT) and do not include the profile information. There can only be a single rdata record specified.

Pool to Pool Conversions

RD Pools of type A (not AAAA) can be converted to and from SiteBacker (SB) and Traffic Controller (TC) pools. To convert between pool types, perform an update (PUT or PATCH) and include the profile information for the appropriate pool type.

RD Pools cannot be converted to or from Directional pools.

TTL Records Consistency in RD Pool Records

Per RFC 2181, the Time to Live (TTL) of all Resource Records in an RRSet must be the same value. Whenever any existing RD Pool record TTL is modified using a partial update request, all other existing RD Pool records will also be updated with the modified TTL value. There will be an audit event for each RD Pool record modification.

Note: Prior to August 15, 2017 it was possible to modify the TTL of any existing RD Pool record using partial update request.

Below is an example scenario of the new TTL function for RD pools.

If there is an existing RD pool with two records:

- Record 1 → Points to 1.1.1.1 with a TTL value of 400
- Record 2 → Points to 1.2.3.4 with a TTL value of 86400

If the TTL value for Record 1 (1.1.1.1) is updated using the partial update request to change the value from 400 to 500, then the additional records will be updated as well. In this scenario, Record 2 (1.2.3.4) will be altered from a TTL value of 86400 to 500.

JSON Example: Update RD pool TTL value

```
{
      "ttl": 500,
      "rdata": [
                 "1.1.1.1"
               ]
}
```

Directional Pools API

The valid RRType values for directional pools are:

- **A**
- **AAAA**
- **PTR**
- **HINFO**
- **MX**
- **TXT**
- **RP**
- **SRV**
- **NAPTR**
- **SPF**

Directional pools can inter-mix CNAME records with either A or AAAA pool types. You cannot specify a directional pool of type CNAME.

Multiple directional pools with the same owner name and type are not allowed.

With the addition of new parameters added to Directional Pools that include GeoIP, Force Overlap, and the TTL v2 update, please review the Profile Fields DTO below carefully to ensure you are using the proper fields in your API calls.

Profile

All pools are implemented as additional information added to *Resource Record Sets* (RRSets). This additional information is specified in a section within the label profile. Every profile contains an entry with the label @context. The value of @context is a URI that uniquely identifies the type of the pool.

The URI for a Directional pool is `http://schemas.ultradns.com/DirPool.jsonschema.`

The other fields in the profile for a Directional Pool are:

Table 33 Directional Pool Profile Fields

Field	Description	Type
description	An optional description of the Directional pool.	String, less than 255 characters. If it is not specified, the owner name for the RRSet will be used.

Field	Description	Type
ignoreECS	Whether to ignore the EDNSO (which is an extended label type allowing for greater DNS message size) Client Subnet data when available in the DNS request. **false** = EDNSO data will be used for IP directional routing. **true** = EDNSO data will not be used and IP directional routing decisions will always use the IP address of the recursive server. Please refer to RFC6891, and RFC7871.	Optional. Boolean. Default value is false. **Note**: Regardless of the value of ignoreECS, conveying the EDNS0 Client Subnet data is not feasible for mixed pools (pools that have both Geo and Source IP selectors). Note: EDNSO data is not used in a directional pool that contains GeoIP and/or Source IP routing.
conflictResolve	When there is a conflict between a matching GeoIP group and a matching SourceIP group, this will determine which should take precedence. This only applies to a mixed pool (contains both GeoIP and SourceIP data).	**GEO** (GeoIP) or **IP** (SourceIP). If not specified, defaults to GEO.
rdataInfo	One entry for each entry in rdata. The order of the rdata entries matches with the order of the rdataInfo entries.	Array of maps of rdataInfo structures.
rdataInfo/allNonConfigured	Indicates whether or not the associated rdata is used for all non-configured geographical territories and SourceIP ranges. At most, one entry in rdataInfo can have this set to true. If this is set to **true**, then geoInfo and ipInfo are ignored.	Boolean – true or false. If not specified, defaults to false.
rdataInfo/geoInfo	The GeoIP group associated with the rdata.	Map of geoInfo structure. Optional.
rdataInfo/geoInfo/name	The name of the GeoIP group.	String. Required.
rdataInfo/geoInfo/isExistingGroupFromPool	Determines if the provided code(s) are already defined in an existing GeoIP group, or if they are a new entry. If True, codes are ignored.	Optional. Default is False. Boolean.

Field	Description	Type
rdataInfo/geoInfo/forceOverlap	Determines the behavior if there is an overlap of GeoIP codes between different records of the same pool. If **true**, the overlapping codes will be removed from conflicting pool level GeoIP groups that have forceOverlap as **false**.	Optional. Default is false. Ignored if isExistingGroupFromPool is True.
rdataInfo/geoInfo/isAccountLevel	**true** if this GeoIP group is referring to an account-level GeoIP group, otherwise **false**. If this is true, codes are ignored.	Boolean. If not specified, defaults to false.
rdataInfo/geoInfo/codes	The codes for the geographical territories that make up this group.	Array of string. See Valid GeoIP Codes for Directional Pools for the valid strings.
rdataInfo/ipInfo	The SourceIP group associated with the rdata.	Map of ipInfo structure. Optional.
rdataInfo/ipInfo/isExistingGroupFromPool	Determines if the provided IPs are already defined in an existing SourceIP group, or if they are a new entry. If **true**, IPs are ignored.	Optional. Default is false. Boolean.
rdataInfo/ipInfo/name	The name of the SourceIP group.	String. Required.
rdataInfo/ipInfo/isAccountLevel	**true** if this SourceIP group is referring to an account-level SourceIP group, otherwise **false**. If this is true, rdataInfo/geoInfo/ codes are ignored.	Boolean. If not specified, defaults to false.
rdataInfo/ipInfo/ips	The list of IP addresses and IP ranges this SourceIP group contains.	Array of IP addresses. Required if this is not an account-level group.
rdataInfo/ipInfo/ips/start	The starting IP address (v4 or v6) for a SourceIP range. If start is present, end must be present as well. CIDR and address must NOT be present.	IPv4 or IPv6 Address.

Field	Description	Type
rdataInfo/ipInfo/ips/end	The ending IP address (v4 or v6) for a SourceIP range. If end is present, start must be present as well. CIDR and address must NOT be present.	IPv4 or IPv6 Address.
rdataInfo/ipInfo/ips/cidr	The CIDR format (IPv4 or IPv6) for an IP address range. If CIDR is present, the start, end, and address must NOT be present.	IPv4 or IPv6 CIDR format Address.
rdataInfo/ipInfo/ips/address	A single IPv4 address. If address is present, the start, end, and CIDR must NOT be present.	IPv4 or IPv6 Address.
rdatainfo/ttl	The Time To Live (in seconds) for the corresponding record in rdata. Must be a value between 0 and 2147483647, inclusive.	Integer.
rdatainfo/type	Returned only on a GET. Indicates the Pool type if the pool type is a subpool. Possible values include: **SLB, SF, SB,** and **TC.** If the Pool type is not a subpool, then the type of pool record will be returned instead.	String.
noResponse	Allows a user to specify certain geographical territories and IP addresses that will get no response if they try to access the directional pool.	Map of rdataInfo. Optional.
noResponse/allNonConfigured	Indicates whether or not “no response” is returned for all non-configured geographical territories and IP ranges. This can only be set to **true** if there is no entry in rdataInfo with allNonConfigured set to true. If this is set to true, then geoInfo and ipInfo are ignored.	Boolean. Optional. Defaults to false.

Field	Description	Type
noResponse/geoInfo	The GeoIP group associated with the “no response” group.	Map of geoInfo structure. Optional.
noResponse/geoInfo/name	The name for the “no response” GeoIP group.	String. Required.
noResponse/geoInfo/isAccountLevel	**true** if the “no response” GeoIP group is referring to an account-level GeoIP group, otherwise **false**. If this is true, codes are ignored.	Boolean. If not specified, defaults to false.
noResponse/geoInfo/codes	The codes for the geographical territories that make up the “no response” group.	An array of string. See *GeoIP Codes for Directional Pools* for the valid strings.
noResponse/ipInfo	The SourceIP group associated with the “no response” group.	Map of IpInfo. Optional.
noResponse/ipInfo/name	The name of the “no response” SourceIP group.	String required.
noResponse/ipInfo/isAccountLevel	**true** if the “no response” SourceIP group is referring to an account-level SourceIP group, otherwise **false**. If this is true, IP addresses are ignored.	Boolean. If not specified, defaults to false.
noResponse/ipInfo/ips	The list of IP addresses and IP ranges for the “no response” SourceIP group.	Array of IP structures. Required if this is not an account-level group.
noResponse/ipInfo/ips/start	The starting IP address (IPv4 or IPv6) for an IP range. If start is present, end must be present as well. CIDR and address must NOT be present.	IPv4 or IPv6 Address.
noResponse/ipInfo/ips/end	The ending IP address (IPv4 or IPv6) for an IP address range. If end is present, start must be present as well. CIDR and address must NOT be present.	IPv4 or IPv6 Address.
noResponse/ipInfo/ips/cidr	The CIDR format (IPv4 or IPv6) for an IP address range. If CIDR is present, start, end, and address must NOT be present.	IPv4 or IPv6 CIDR format Address.

Field	Description	Type
noResponse/ipInfo/ips/address	A single IPv4 or IPv6 address. If an IP address is present, start, end, and CIDR must NOT be present.	IPv4 or IPv6 Address.

JSON Example: Directional Pool RRSet with Profile

```
{
    "zoneName": "domain.name.",
    "ownerName": "pool.domain.name.",
    "rrtype": "A",
    "ttl": 300,
    "type":
    "rdata": [
        "1.2.3.4",
        "a.domain.name.",
        "9.8.7.6",
        "30.40.50.60"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
        "description": "Description of pool",
        "conflictResolve": "GEO",
        "ignoreECS": true,
        "rdataInfo": [
            {
                "allNonConfigured": true
            },
            {
                "geoInfo": {
                    "name": "North America",
                    "codes": [ "US", "CA", "MX"]
                }
            },
            {
                "ipInfo": {
                    "name": "some Ips",
                    "ips": [
                        {
                            "start": "200.20.0.1",
                            "end": "200.20.0.10"
                        },
                        {
                            "cidr": "20.20.20.0/24"
                        },
                        {
                            "address": "50.60.70.80"
                        }
                    ]
                }
            },
            {
                "geoInfo": {
```

```
                    "name": "accountGeoGroup",
                    "isAccountLevel": true
                },
                "ipInfo": {
                    "name": "accountIPGroup",
                    "isAccountLevel": true
                }
            }
        ],
        "noResponse": {
            "geoInfo": {
                "name": "nrGeo",
                "codes": ["Z4"]
            },
            "ipInfo": {
                "name": "nrIP",
                "ips": [
                    {
                        "address": "197.231.41.3"
                    }
                ]
            }
        }
}
```

JSON Example: Add Existing Global Group to a Directional Pool

The following JSON example outlines how a Global Directional Group can be added to a Directional Pool for more than one record. To do this, the local group name must be provided in the group name, and the **isAccountLevel** and **isExistingGroupFromPool** parameters must be set to true. In the following example, first record is the allNonConfigured record, the second record is assigned to global group "groupName," and the third record is being assigned the same global group again in the pool.

```
{
    "ownerName": "pool-zone.com.",
    "rrtype": "SRV (33)",
    "rdata": [
        "1 11 12 target1.com",
        "2 2 2 target2.com",
        "3 2 2 target3.com"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
        "description": "srv",
        "rdataInfo": [
            {
                "allNonConfigured": true,
                "ttl": 86400,
                "type": "SRV"
            },
            {
                "geoInfo": {
                    "name": "groupName",
```

```
                    "isAccountLevel": true
                },
                "ttl": 86400,
                "type": "SRV"
            },
            {
                "geoInfo": {
                    "name": "groupName",
                    "isAccountLevel": true,
                    "isExistingGroupFromPool":true
                },
                "ttl": 86400,
                "type": "SRV"
            }
        ]
    }
}
```

Configuring GeoIP and Source IP Together

Depending on your needs to configure your directional pool records, you may opt to combine a Source IP group and a GeoIP group together into one directional record. When you do so, the record gets selected only when **BOTH** the Source IP **and** GeoIP restrictions are satisfied. For example:

```
POST
{
    "ownerName": "and.000geoiptest.net.",
    "rrtype": "A (1)",
    "rdata": [
                "1.1.1.1",
                "2.2.2.2"
             ],
     "profile": {
                 "@context": "http://schemas.ultradns.com/DirPool.jsonsc
hema",
                 "rdataInfo": [
                    {
                   "allNonConfigured": true
                   },
                   {
                     "geoInfo": {
                         "name": "br",
                         "codes": [
                                   "BR"
                                   ]
                                 },
                      "ipInfo": {
                      "name": "some Ips",
                      "ips": [
                                {
                      "start": "200.20.0.1",
                      "end":   "200.20.0.10"
                               }
                              ]
```

```
                                    }
                                }
                            ]
                        }
}
```

The directional record 2.2.2.2 above will be selected **only** if you are in Brazil, **AND** your Source IP is within the range [200.20.0.1 – 200.20.0.10]. However, if you were coming from Brazil, but your Source IP was 200.20.0.55 (outside of the range), the All Non-Configured resolution of 1.1.1.1 would be chosen.

Similarly, if using the above example, the Source IP range of 200.20.0.1 – 200.20.0.10 no longer resided in Brazil, and the user's IP Address was coming from 200.20.0.1, the resolution would once again choose the All Non-Configured record of 1.1.1.1 because you did not satisfy both the GeoIP and Source IP restrictions.

If you continue to receive the All Non-Configured response when combining GeoIP and Source IP, verify you satisfy both of the restrictions.

TTL Update

The Time to Live (TTL) for Directional Pools can optionally be set at the record level via PUT, POST and PATCH calls. The GET call will return record level TTL values for Directional Pools.

Sub Pool TTLs cannot be updated using the TTL optional field.

The priority of the TTL display will act as follows for POST, PUT and PATCH calls:

- If provided, the Record level TTL will take priority over the Pool TTL.
- If no record TTL is provided, the Pool level TTL will be used for both Record and Pool TTLs.

NOTE: The RESTAPI does not validate the compatibility of a pool or a pool record's TTL versus that of a sub-pool or a sub-pool's record TTL. A sub-pool's TTL value will take precedence over the controlling (parent) pool's TTL value.

For example, if you were to create a Directional pool named dir.example.com that contained a record named sb.example.com which points to a SiteBacker pool, then the TTL for the Sitebacker pool records will take precedence and be used upon resolution of the dir.example.com pool. The TTL(s) of the Directional pool records will be ignored upon resolution, but still accessible and able to be updated by the REST API.

Below are various JSON examples depicting the hierarchy of TTLs per record type / pool type.

JSON Example: Determining the TTL of a pool record

Step 1: Create (POST) a SiteBacker Pool with TTL of 180

```
{
      "ownerName": "sb.000geoiptest.net.",
```

```
      "rrtype": "A (1)",
      "ttl": 180,
      "rdata": [
      "1.1.1.1"
        ],
          "profile": {
              "@context": "http://schemas.ultradns.com/SBPool.jsonschema",
              "description": "sb.000geoiptest.net.",
              "runProbes": true,
              "actOnProbes": true,
              "order": "ROUND_ROBIN",
              "maxActive": 1,
              "maxServed": 0,
              "rdataInfo": [
                  {
                      "state": "NORMAL",
                      "runProbes": true,
                      "priority": 1,
                      "failoverDelay": 0,
                      "threshold": 1
                  }
           ]
}
```


Step 2: Create (POST) a Directional Pool with the previous SiteBacker Pool as a Subpool, and the Directional Pool has a TTL of 1500.


```
{
     "ownerName": "dir.000geoiptest.net.",
     "rrtype": "A (1)",
     "rdata": [
              "sb.000geoiptest.net."
               ],
          "profile": {
              "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
              "description": "dir.000geoiptest.net.",
              "rdataInfo": [
                   {
                      "geoInfo": {
                          "name": "na",
                          "codes": [
                             "NAM"
                            ]
                        },
                        "ttl": 1500
                    }
                ]
            }
        }
```


The "dir.000geoiptest.net" resolves to 1.1.1.1 and with a TTL of 180.

Step 3: Perform a GET of the SiteBacker Pool shows the TTL of 180.


```
{
```

```
    "zoneName": "000geoiptest.net",
    "rrSets": [
        {
            "ownerName": "sb.000geoiptest.net.",
            "rrtype": "A (1)",
            "ttl": 180,
            "rdata": [
                "1.1.1.1"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/SBPool.jsonschema",
                "description": "sb.000geoiptest.net.",
                "runProbes": true,
                "actOnProbes": true,
                "order": "ROUND_ROBIN",
                "maxActive": 1,
                "maxServed": 0,
                "rdataInfo": [
                    {
                        "state": "NORMAL",
                        "runProbes": true,
                        "priority": 1,
                        "failoverDelay": 0,
                        "threshold": 1,
                        "availableToServe": true,
                    }
                ]
            }
        }
    ],
```


Step 4: Perform a GET of the Directional Pool shows the TTL of 1500.


```
{
    "zoneName": "000geoiptest.net",
    "rrSets": [
        {
            "ownerName": "dir.000geoiptest.net.",
            "rrtype": "A (1)",
            "rdata": [
                "sb.000geoiptest.net."
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "dir.000geoiptest.net.",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "na",
                            "codes": [
                                "NAM"
                            ]
                        },
                        "ttl": 1500,
                        "type": "SB"
                    }
```

```
                ]
            }
        }
    ],
```

Step 5: Update (PUT) the TTL of the Directional Pool to 1600.

```
{
            "ownerName": "dir.000geoiptest.net.",
            "rrtype": "A (1)",
            "rdata": [
                "sb.000geoiptest.net."
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "rdataInfo": [
                    {
                        "ttl": 1600
                    }
                ]
            }
}
```

Step 6: Update (PUT) the TTL of the SiteBacker Pool to 200.

```
{
     "ownerName": "sb.000geoiptest.net.",
     "rrtype": "A (1)",
     "ttl": 200,
     "profile": {
             "@context": "http://schemas.ultradns.com/SBPool.jsonschema"
            }
        }
```

The "sb.000geoiptest.net" resolves with a TTL of 200.
Then the "dir.000geoiptest.net" resolves with a TTL of 200.

Step 7: Perform a GET of the SiteBacker Pool shows a TTL of 200.

```
{
    "zoneName": "000geoiptest.net",
    "rrSets": [
        {
            "ownerName": "sb.000geoiptest.net.",
            "rrtype": "A (1)",
            "ttl": 200,
            "type": SB,
            "rdata": [
                "1.1.1.1"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/SBPool.jsonschema",
                "description": "sb.000geoiptest.net.",
                "runProbes": true,
                "actOnProbes": true,
```

```
                "order": "ROUND_ROBIN",
                "maxActive": 1,
                "maxServed": 0,
                "rdataInfo": [
                    {
                        "state": "NORMAL",
                        "runProbes": true,
                        "priority": 1,
                        "failoverDelay": 0,
                        "threshold": 1,
                        "availableToServe": true
                    }
                ]
            }
        }
    ],
```


Step 8: Perform a GET of the Directional Pool shows a TTL of 1800.


```
{
    "zoneName": "000geoiptest.net",
    "rrSets": [
        {
            "ownerName": "dir.000geoiptest.net.",
            "rrtype": "A (1)",
            "rdata": [
                "sb.000geoiptest.net."
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "dir.000geoiptest.net.",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "na",
                            "codes": [
                                "NAM"
                            ]
                        },
                        "ttl": 1600,
                        "type": "SB"
                    }
                ]
            }
        }
    ],
```


The above steps are used as an example to depict the differences in how a TTL will be displayed / applied to a pool, which take into consideration whether the pool is a subpool, and in what order the records are resolved.

The following examples demonstrate the difference between the /v1/ and the current version usages in the REST API.

JSON Example: REST API /v1/ TTL Response only at the Pool Level


```
"zoneName": "restapi.test.com",
    "rrSets": [
        {
            "ownerName": "ttl.test.biz",
            "rrtype": "A (1)",
            "ttl": 1000,
            "type":
            "rdata": [
                "212.82.0.6"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "ttl.test.biz",
                "conflictResolve": "GEO",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "g42",
                            "codes": [
                                "AR",
                                "BO",
                                "BR",
                                "CL",
                                "CO",
                                "EC",
                                "FK",
                                "GF",
                                "GS",
                                "GY",
                                "PE",
                                "PY",
                                "SR",
                                "U4",
                                "UY",
                                "VE"
                            ]
                        },
                        "type": "A"
                    }
                ]
            }
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```

JSON Example: REST API Current version TTL Response at Record Level


```
"zoneName": "restapi.test.com",
    "rrSets": [
        {
            "ownerName": "ttl.test.biz",
            "rrtype": "A (1)",
            "rdata": [
                "212.82.0.6"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "ttl.test.biz",
                "conflictResolve": "GEO",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "g42",
                            "codes": [
                                "SAM"
                            ]
                        },
                        "ttl": 1000,
                        "type": "A"
                    }
                ]
            }
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```


Force Overlap

Force Overlap is a parameter that is only applicable for GeoIP data. This parameter determines the behavior if there is an overlap of GeoIP codes between different records of the same pool. If the value is True, the overlapping codes will be removed from conflicting pool level GeoIP groups that have forceOverlap set to False.

For example, if pool record A is existing and set up for “Asia” and “North America,” and record B is added to the same pool with “North America” and “Africa” with forceOverlap set to True, the resulting pool will display “Asia” for record A, and “North America” and “Africa” for record B.

Create a Directional Pool

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must contain *Directional Pool Profile Fields*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.

JSON Example: Create Directional Pool with GeoIP location data

```
{
      "ownerName": "Myaccount",
      "rdata": [
                "txt1","txt2"
            ],
         "profile": {
         "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
         "rdataInfo": [
               {
                     "geoInfo": {
                     "name": "gr1", "codes": ["ASI", "AFR"]
                      }
                   },
                   {
                    "geoInfo": {
                       "name":"gr1",
                       "isExistingGroupFromPool": true,
                       "codes":["ANT"]
                      }
                 }
            ]
      }
}
```

JSON Example: Create Directional Pool with ignoreECS flag enabled

```
{
      "ownerName": "dir-ignoreECS",
      "rdata": [
          "txt1","txt2"
      ],
      "profile": {
         "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
```

```
        "ignoreECS": true,
        "rdataInfo": [
        {
            "geoInfo": {
                "name": "gr1", "codes": ["ASI", "AFR"]
             }
        },
        {
            "geoInfo": {
                 "name":"gr1",
                 "isExistingGroupFromPool": true,
                 "codes":["ANT"]
            }
        }
      ]
   }
}
```

Get a Directional Pool

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/TXT/{ownerName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with *Profile DTO* details in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.

JSON Example: Get a Directional Pool with Force Overlap enabled

```
{
    "zoneName": "ForceOverlap.biz",
    "rrSets": [
        {
            "ownerName": "MyAccount",
            "rrtype": "TXT (16)",
            "type:"
            "rdata": [
                "\"txt1\"",
                "\"txt2\""
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "MyAccount",
                "rdataInfo": [
                    {
```

```
                        "geoInfo": {
                            "name": "gr1",
                            "codes": [
                                "AFR",
                                "ASI"
                            ]
                        },
                        "ttl": 86400,
                        "type": "TXT"
                    },
                    {
                        "geoInfo": {
                            "name": "gr1",
                            "codes": [
                                "AFR",
                                "ASI"
                            ]
                        },
```

JSON Example: Get a Directional Pool with ignoreECS flag enabled

```
{
    "zoneName": "NewDirPool.biz",
    "rrSets": [
        {
            "ownerName": "dir-ignoreECS",
            "rrtype": "TXT (16)",
            "type:"
            "rdata": [
                "txt1",
                "txt2"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "dir-ignoreECS.NewDirPool.biz",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "gr1",
                            "codes": [
                                "AFR",
                                "ASI"
                            ]
                        },
                        "ttl": 3900,
                        "type": "TXT"
                    },
                    {
                        "geoInfo": {
                            "name": "gr2",
                            "codes": [
                                "ANT",
                                "EUR"
                            ]
                        },
                    "ttl": 3900,
```

```
                    "type": "TXT"
                }
            ]
        }
    }
],
"queryInfo": {
    "sort": "OWNER",
    "reverse": false,
    "limit": 100
},
"resultInfo": {
    "totalCount": 1,
    "offset": 0,
    "returnedCount": 1
}
}
```


Get All Directional Pools

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:DIR_POOLS
```

Parameters: None

Body: None.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.

Users should provide a maximum limit of 1000 when requesting to get all Directional pools. If the limit is greater than 1000, the following error message will be returned:


```
{
errorCode: 22000
errorMessage: "Invalid Page Limit, the maximum number of records that can be
retrieved are restricted to 1000."
}
```


In order to acquire more than 1,000 records (if necessary), you can change the offset and limit parameters to satisfy the record range you are seeking. For example, to get records 2,001 through 3,000 change offset=2000&limit=1000.

Partially Update a Directional Pool

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with a Directional pool profile info, or a *JSON PATCH DTO*.

Patchable Objects for DIR Pool:

- biz.neustar.ultra.rest.dto.DirectionalPoolProfile

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don’t have permission to partial update DIR Pools.
- If patch operation is not allowed for the given json pointer value.

JSON Example: Partial Update a Directional Pool with GeoIP location enabled

```
{
      "ownerName": "ForceOverlap.biz",
      "rdata": [
              "txt2"
               ],
           "profile": {
              "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
              "rdataInfo": [
                    {
                   "geoInfo": {
                      "name":"gr2", "codes":["ANT", "EUR", "ASI"],
                      "forceOverlap": true
                      }
                  }
            ]
      }
}
```

JSON Example: Partial Update a Directional Pool with ignoreECS flag enabled

```
{
      "ownerName": "dir-ignoreECS",
      "rdata": [
          "txt2"
      ],
      "profile": {
```

```
          "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
          "ignoreECS": false,
          "rdataInfo": [
          {
              "geoInfo": {
                  "name":"gr2", "codes":["ANT", "EUR", "ASI"],
                  "forceOverlap": true
               }
          }
      ]
   }
}
```

Update Directional Pools

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with a Directional pool profile info, or a *JSON PATCH DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don’t have permission to partial update DIR Pools.

Users can delete a specific record from a DIR pool by sending a PUT request without providing that record in the input.

JSON Example: Update Directional Pool with GeoIP location

```
{
         "ownerName": ForceOverlap.biz",
            "rdata": [
                "txt1","txt2"
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "rdataInfo": [
               {
                      "geoInfo": {
                        "name": "gr1", "codes": ["ASI", "AFR"]
                      }
                   },
```

```
                    {
                      "geoInfo": {
                        "name":"gr2", "codes":["ANT", "EUR"]
                      }
                    }
            ]
      }
}
```

JSON Example: Update Directional Pool with ignoreECS flag

```
{
      "ownerName": "dir-ignoreECS",
      "rdata": [
          "txt1","txt2"
      ],
      "profile": {
          "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
          "ignoreECS": true,
          "rdataInfo": [
          {
              "geoInfo": {
                  "name": "gr1", "codes": ["ASI", "AFR"]
              }
          },
          {
              "geoInfo": {
                  "name":"gr2", "codes":["ANT", "EUR"]
              }
          }
      ]
   }
}
```

Valid GeoIP Codes for Directional Pools

The REST API follows the ISO-3661-1 standard to specify the names of countries. REST API is using the Alpha-2 (two letter) code format. The ISO-3166-2:US standard is used to represent US states and territories, and the ISO-3166-2:CA standard is used to represent Canadian provinces and territories.

Any geographic location can be included in any directional pool record. The pool record with the most specific geographic location will take precedence when matching the IP address to the location. The State / Province takes precedence over country, which takes precedence over continent. For example, an IP address originating in Virginia, USA, and North America would select the “Virginia” record over the USA or North America location. If there is no “Virginia” record defined, the next most specific match would then be USA.

For a comprehensive list of sub-country codes recognized by the REST API, please refer to the *Geo-IP ISO Code Guide* (which can also be found on the Support page of the Ultradns Portal), or use the *GET GeoIP Territories* **(geoip/territories)** end point. In most instances, the sub country codes recognized by the REST API for a given country (XX) will be a full set, or a sub set of the ISO-3166-2:XX standard for that country.

REST API will utilize localized spelling for State/Province names. For example, the state of Tuscany in Italy will be identified as “Toscana.”

In addition, other non-standard codes have been defined to represent other geographical areas, as well as DNS records that don't fall into geographical territories:

Table 34 GeoIP Codes for Directional Pools

Code	Meaning	Equivalent ISO codes
A1	Anonymous Proxy	None
A2	Satellite Provider	None
A3	Unknown / Uncategorized IPs	None
NAM	North America (including Central America and the Caribbean)	AG,AI,AN,AW,BB,BL,BM,BQ,BS,BZ,CA,CR,CU,CW,DM,DO,GD,GL,GP,GT,HN,HT,JM,KN,KY,LC,MF,MQ,MS,MX,NI,PA,PM,PR,SV,SX,TC,TT,U3,US,VC,VG,VI
SAM	South America	AR,BO,BR,CL,CO,EC,FK,GF,GS,GY,PE,PY,SR,U4,UY,VE
EUR	Europe	AD,AL,AM,AT,AX,AZ,BA,BE,BG,BY,CH,CZ,DE,DK,EE,ES,FI,FO,FR,GB,GE,GG,GI,GR,HR,HU,IE,IM,IS,IT,JE,LI,LT,LU,LV,MC,MD,ME,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SJ,SK,SM,U5,UA,VA
AFR	Africa	AO,BF,BI,BJ,BW,CD,CF,CG,CI,CM,CV,DJ,DZ,EG,EH,ER,ET,GA,GH,GM,GN,GQ,GW,KE,KM,LR,LS,LY,MA,MG,ML,MR,MU,MW,MZ,NA,NE,NG,RE,RW,SC,SD,SH,SL,SN,SO,SS,ST,SZ,TD,TG,TN,TZ,U7,UG,YT,ZA,ZM,ZW
ASI	Asia (including Middle East and the Russian Federation)	AE,AF,BD,BH,BN,BT,CN,CY,HK,ID,IL,IN,IO,IQ,IR,JO,JP,KG,KH,KP,KR,KW,KZ,LA,LB,LK,MM,MN,MO,MV,MY,NP,OM,PH,PK,PS,QA,RU,SA,SG,SY,TH,TJ,TL,TM,TR,TW,U6,U8,UZ,VN,YE
OCN	Australia / Oceania	AS,AU,CC,CK,CX,FJ,FM,GU,HM,KI,MH,MP,NC,NF,NR,NU,NZ,PF,PG,PN,PW,SB,TK,TO,TV,U9,UM,VU,WF,WS
ANT	Antarctica	AQ, TF, BV
Legacy Codes		
US-U1	Undefined United States	
CA-U2	Undefined Canada	
U3	Undefined Central America	
U4	Undefined South America	
U5	Undefined Europe	
U6	Undefined Middle East	
U7	Undefined Africa	
U8	Undefined Asia	

Code	Meaning	Equivalent ISO codes
U9	Undefined Australia / Oceania	
Z1	The Caribbean	AI, AG, AW, BS, BB, BM, VG, KY, CU, DM, DO, GD, GP, HT, JM, MQ, MS, AN, PR, BL, MF, VC, KN, LC, TT, TC, VI
Z2	Central America	BZ, CR, SV, GT, HN, NI, PA, U3
Z3	South America	AR, BO, BR, CL, CO, EC, FK, GF, GY, PY, PE, GS, SR, U4, UY, VE
Z4	Europe	AX, AL, AD, AM, AT, AZ, BY, BE, BA, BG, HR, CZ, DK, EE, FO, FI, FR, GE, DE, GI, GR, GG, HU, IS, IE, IM, IT, JE, LV, LI, LT, LU, MK, MT, MD, MC, ME, NL, NO, PL, PT, RO, SM, RS, SK, SI, ES, SJ, SE, CH, UA, U5, GB, VA
Z5	Middle East	AF, BH, CY, IR, IQ, IL, JO, KW, LB, OM, PS, QA, SA, SY, TR, U6, AE, YE
Z6	Africa	DZ, AO, BJ, BW, BF, BI, CM, CV, CF, TD, KM, CG, CI, CD, DJ, EG, GQ, ER, ET, GA, GM, GH, GN, GW, KE, LS, LR, LY, MG, MW, ML, MR, MU, YT, MA, MZ, NA, NE, NG, RE, RW, ST, SN, SC, SL, SO, ZA, SH, SD, SZ, TZ, TG, TN, UG, U7, EH, ZM, ZW
Z7	Asia (Excluding Middle East and the Russian Federation)	BD, BT, IO, BN, KH, CN, HK, IN, ID, JP, KZ, KP, KR, KG, LA, MO, MY, MV, MN, MM, NP, PK, PH, SG, LK, TW, TJ, TH, TL, TM, U8, UZ, VN
Z8	Australia / Oceania	AS, AU, CX, CC, CK, FJ, PF, GU, HM, KI, MH, FM, NR, NC, NZ, NU, NF, MP, PW, PG, PN, WS, SB, TK, TO, TV, U9, VU, WF
Z9	Antarctica	AQ, TF, BV

Using the ISO code US is equivalent to all the codes in ISO-3166-2:US, except for the outlying territories which include American Samoa, Guam, Northern Mariana Islands, Puerto Rico, and the Virgin Islands, plus US-U1.In addition to the ISO-3166-2:US codes, there are 3 Armed Forces Designations included: US-AA, US-AE, and US-AP. Similarly, the ISO code CA is equivalent to ISO-3166-2:CA plus CA-U2.

To ensure the backward compatibility of existing customer verification scripts, GeoIP Codes returned by a GET /v1 endpoint will match the behavior of the REST API before the August 2016 update. Customers using the /v1/ GeoIP Codes are encouraged to switch to the new code format.

Table 35 Deprecated ISO Codes below contains GeoIP Codes for which the past behavior of the REST API is preserved in the /v1 versions of the corresponding GET endpoints. For example, if you create a directional record for South America, the result returned by the corresponding /v1 GET endpoint would list all of the eight country codes associated with South America, as opposed to a single GeoIP code “SAM” that corresponds to the continent of South America as a whole.

Table 35 Deprecated ISO Codes

ISO Code	Description
US	United States
CA	Canada
Z3 / SAM	South America
Z4 / EUR	Europe
Z6 / AFR	Africa
Z8 / OCN	Australia
Z9 / ANT	Antarctica
GL	Greenland
PM	Saint Pierre and Miquelon

JSON Example: /v1/ ISO Code Example

```
"profile": {
                "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
                "description": "restapi.test.com",
                "conflictResolve": "GEO",
                "rdataInfo": [
                    {
                        "geoInfo": {
                            "name": "g42",
                            "codes": [
                                "AR",
                                "BO",
                                "BR",
                                "CL",
                                "CO",
                                "EC",
                                "FK",
                                "GF",
                                "GS",
                                "GY",
                                "PE",
                                "PY",
                                "SR",
                                "U4",
                                "UY",
                                "VE"
                            ]
                    }
                }
      ]
}
```

JSON Example: Current version ISO Code Example

```
"profile": {
          "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
          "description": "restapi.test.com",
          "conflictResolve": "GEO",
          "rdataInfo": [
```

```
                    {
                     "geoInfo": {
                           "name": "g42",
                           "codes": [
                               "SAM"
                                ]
                             },
                       }
                  ]
          }
]
```

Parent / Child Territories Overlap

With the rollout of the Global State Level GeoIP feature, any geographic location can be included in any directional pool record. The pool record with the most specific geographic location will take precedence when matching the IP address to the correlating location.

Previously, it would not have been possible to have the state of Virginia (US-VA) in one record, and the country of United States (US) in another record in the same pool. The system would have forced the exclusion of the state of Virginia from the second record, effectively replacing the United States (US) with the list of remaining states (all excluding Virginia).

JSON Example: Parent / Child Territory Overlap

```
POST /zones/{zoneName}/rrsets/A/{poolName}
{
    "rdata": ["1.1.1.1","1.1.1.2","1.1.1.3"],
    "profile": {
        "@context": "http://schemas.ultradns.com/DirPool.jsonschema",
        "rdataInfo": [
            {"geoInfo": {"name": "g1", "codes":["MX-MEX"]}},
            {"geoInfo": {"name": "g2", "codes":["MX"]}},
            {"geoInfo": {"name": "g3", "codes":["NAM"]}}
                    ]
               }
}
```

The above JSON Example demonstrates that if the source IP is in the ‘State’ of Mexico, the resolution is 1.1.1.1. If the source IP is in any other part of the country of Mexico, the resolution is 1.1.1.2. If the source IP is in any other part in North America, the resolution is 1.1.1.3. In this example, there is no overlap.

The new GeoIP model will allow for various Parent/Child overlaps. You can now have a record with the state of Virginia, another record for the U.S., and still another record for all of North America.

GET GeoIP Territories

```
GET  https://api.ultradns.com/geoip/territories
```

Parameters:

Parameter	Description	Type
codes	Comma-separated list of one or more territories. You can pass an empty string for a top level return (continents only returned). You can pass a three-letter continent code (refer to the spreadsheet for a complete list of GeoIP ISO codes) for a continent. You can drill down further by passing a continent code dash separated (-) ISO-3166 country code for a country (NAM-MX).	String.

Territory DTO

Table 36 GeoIP Territory DTO

Field	Description	Type
name	Name of the territory.	String.
code	GeoIP Code assigned to the territory. (Reference Excel Spreadsheet)	String.
type	One of the following: Territory (Continent), Country, State.	String.
id	Integer used for internal numeric id to represent the territory.	Integer.

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Territory DTO* elements for each comma-separated code from the input parameter in the body content.

Errors: An error will be returned under the following conditions:

- A duplicate Continent/Country code was provided.
- The return contains more than 100 records.
- The ISO code provided is a non-existent ISO code.

JSON Example: GeoIP Current version Territory Empty String Return

```
GET https://api.ultradns.com/geoip/territories?codes=
{
            "name": "Anonymous Proxy",
            "code": "A1",
            "type": "Country",
            "id": 315
        },
        {
            "name": "Satellite Provider",
            "code": "A2",
            "type": "Country",
            "id": 316
        },
        {
            "name": "Unknown / Uncategorized IPs",
            "code": "A3",
            "type": "Country",
            "id": 331
        },
```

```
        {
            "name": "North America",
            "code": "NAM",
            "type": "Region",
            "id": 338
        },
        {
            "name": "South America",
            "code": "SAM",
            "type": "Region",
            "id": 337
        },
        {
            "name": "Europe",
            "code": "EUR",
            "type": "Region",
            "id": 336
        },
        {
            "name": "Africa",
            "code": "AFR",
            "type": "Region",
            "id": 332
        },
        {
            "name": "Asia",
            "code": "ASI",
            "type": "Region",
            "id": 334
        },
        {
            "name": "Australia / Oceania",
            "code": "OCN",
            "type": "Region",
            "id": 335
        },
        {
            "name": "Antarctica",
            "code": "ANT",
            "type": "Region",
            "id": 333
        }
    ]
]
```

JSON Example: GeoIP Current version with 2 Territories comma-separated

```
GET https://api.ultradns.com/geoip/territories?codes=ANT,EUR-AD
[
    [
        {
            "name": "Antarctica",
            "code": "AQ",
            "type": "Country",
            "id": 330
        },
```

```
        {
            "name": "Bouvet Island",
            "code": "BV",
            "type": "Country",
            "id": 297
        },
        {
            "name": "French Southern Territories",
            "code": "TF",
            "type": "Country",
            "id": 298
        }
    ],
    [
        {
            "name": "Andorra La Vella",
            "code": "07",
            "type": "State",
            "id": 351
        },
        {
            "name": "Canillo",
            "code": "02",
            "type": "State",
            "id": 346
        },
        {
            "name": "Encamp",
            "code": "03",
            "type": "State",
            "id": 347
        },
        {
            "name": "Escaldes-Engordany",
            "code": "08",
            "type": "State",
            "id": 352
        },
        {
            "name": "La Massana",
            "code": "04",
            "type": "State",
            "id": 348
        },
        {
            "name": "Ordino",
            "code": "05",
            "type": "State",
            "id": 349
        },
        {
            "name": "Sant Julia De Loria",
            "code": "06",
            "type": "State",
            "id": 350
        }
```

```
    ]
]
```

Get All Account-level GeoIP Groups

Retrieve a list of all configured account-level GeoIP directional groups for a specified GeoIP code, for a particular account.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/dirgroups/geo
```

Parameters:

Table 37 GeoIP AccountList Parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are name, which is a substring match for any GeoIP group name.	String.
offset	The position in the list for the first returned element (0 based). The default value is “0”.	Integer.
limit	The maximum number of rows requested. The default value is 100.	Integer.
sort	The sort column used to order the list. Valid values are NAME.	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). The default value is false.	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Account-Level GeoIP Directional Group List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to read account-level groups.

Get All Account-level SourceIP Groups

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/dirgroups/ip
```

Parameters:

Table 38 IP AccountList Parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are name, which is a substring match for any IP group name.	String.
offset	The position in the list for the first returned element (0 based). The default value is “0”.	Integer.

Parameter	Description	Type
limit	The maximum number of rows requested. The default value is “100”.	Integer.
sort	The sort column used to order the list. Valid values are “NAME”.	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). The default value is false.	Boolean.

Body: None

Response: If task completes, Status Code 200 is returned with an *Account-Level IP Directional Group List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to read account-level groups.

Get an Account-level GeoIP Group

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/dirgroups/geo/{name}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Account-Level GeoIP Directional Group DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to read account-level directional groups.
- If {name} is not the name of an account-level GeoIP directional group.

Get an Account-level SourceIP Group

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/dirgroups/ip/{name}
```

Parameters: None.

Body: None.

Response: If task completes, Status Code 200 is returned with an *Account-Level SourceIP Directional Group DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to read account-level directional groups.
- If {name} is not the name of an account-level IP directional group.

Create an Account-level GeoIP Group

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/dirgroups/geo/{name}
```

Parameters: None

Body: Must include an *Account-Level GeoIP Directional Group DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to create account-level geo group.

Create an Account-level SourceIP Group

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/dirgroups/ip/{name}
```

Parameters: None.

Body: Must contain an *Account-Level SourceIP Directional Group DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to create account-level IP group.

Update an Account-level GeoIP Group

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/dirgroups/geo/{name}
```

Parameters: None

Body: Must include an *Account-Level GeoIP Directional Group DTO* in the body content.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to edit account-level directional groups.
- If {name} is not the name of an account-level Geo directional group.

Update an Account-level SourceIP Group

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/dirgroups/ip/{name}
```

Parameters: None.

Body: Must contain an *Account-Level SourceIP Directional Group DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to edit account-level directional groups.
- If {name} is not the name of an account-level IP directional group.

Partially Update an Account-level GeoIP Group

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/dirgroups/geo/{name}
```

Parameters: None

Body: Must include an *Account-Level GeoIP Directional Group DTO* in the body content.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to edit account-level directional groups.
- If {name} is not the name of an account-level Geo directional group.

Partially Update an Account-level SourceIP Group

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/dirgroups/ip/{name}
```

Parameters: None.

Body: Must contain an *Account-Level SourceIP Directional Group DTO* or a *JSON PATCH DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to edit account-level directional groups.
- If {name} is not the name of an account-level IP directional group.

JSON Example: Partially Updating an Account-Level SourceIP Group with JSON PATCH

```
[
   {
    "body": [
        {
            "op": "remove",
            "path": "/ips/4.4.4.4%2F32"
        }
    ],
    "method": "PATCH",
    "uri": "/accounts/netflix/dirgroups/ip/foo1"
}
,
{
    "body": [
        {
            "op": "remove",
            "path": "/ips/2.2.2.2%2F32"
        }
    ],
    "method": "PATCH",
    "uri": "/accounts/netflix/dirgroups/ip/foo2"
},
{
```

```
    "body": [
        {
            "op": "add",
            "path": "/ips/2.2.2.2%2F32"
        }
    ],
    "method": "PATCH",
    "uri": "/accounts/netflix/dirgroups/ip/foo1"
}
,
{
    "body": [
        {
            "op": "add",
            "path": "/ips/4.4.4.4%2F32"
        }
    ],
    "method": "PATCH",
    "uri": "/accounts/netflix/dirgroups/ip/foo2"
}
]
```

Delete an Account-level GeoIP Group

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/dirgroups/geo/{name}
```

Parameters: None.

Body: None.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to delete account-level directional groups.
- If {name} is already in use by another account-level directional group (either IP or Geo).

Delete an Account-level SourceIP Group

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/dirgroups/ip/{name}
```

Parameters: None.

Body: None.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {accountName} is not valid, or not an {accountName} you have access to.
- If you don't have permission to delete account-level directional groups.
- If {name} is already in use by another account-level directional group (either IP or Geo).

Account Level Directional Group DTOs

Account-Level GeoIP Directional Group DTO

Table 39 Account-Level GeoIP Group DTO Structure

Field	Description	Type
name	The name of the group.	String. Required in GET response, ignored in POST/PATCH/PUT, since it's specified in the URI.
description	The description for the group.	String. Optional

JSON Example: Account-Level GeoIP Group (in context)

```
{
    "name": "accountGeoGroup",
    "description": "A sample group",
    "codes": ["Z6", "RU"]
}
```

Account-Level SourceIP Directional Group DTO

Table 40 Account-Level SourceIP Group DTO Structure

Field	Description	Type
name	The name of the group.	String. Required for GET; ignored in POST/PATCH/PUT.
description	The description for the group.	String. Optional.
ips	The list of IP addresses this group contains.	Array of IP structures.
ips/start	The starting IP address (v4) for an IP range. If start is present, end must be present as well. CIDR and address must NOT be present	IPv4
ips/end	The ending IP address (v4) for an IP range. If end is present, start must be present as well. CIDR and address must NOT be present.	IPv4
ips/cidr	The CIDR format (v4) for an IP address range. If CIDR is present, start, end, and address must NOT be present.	IPv4 CIDR format.

Field	Description	Type
ips/address	A single IPv4 address. If address is present, start, end, and CIDR must NOT be present.	IPv4

JSON Example: Account-Level SourceIP Group (in context)

```
{
    "name": "accountIPGroup",
    "description": "Another sample",
    "ips": [
        {
            "start": "1.1.1.1",
            "end": "2.2.2.2"
        },
        {
            "address": "4.3.2.1"
        }
    ]
}
```

Account-Level GeoIP Directional Group List DTO

This is returned when retrieving multiple Account-Level GeoIP Groups from the server. It is not used for creating or updating Account-Level GeoIP Groups.

Table 41 Account-Level GeoIP Group List DTO Structure

Field	Description	Type
accountName	The account name for the groups.	String.
geoGroups	The list of all Account-Level GeoIP Groups that matched the query for the offset and limit.	List of *Account-Level GeoIP Directional Group DTOs.*
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all events in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: Account-Level GeoIP Group List

```
{
    "accountName" : "myAccount" ,
    "geoGroups" : [
        {
            "name": "accountGeoGroup",
```

```
            "description": "geo sample",
            "codes": [ "FR", "GB", "RU" ]
        },
        {
            "name": "accountGeoGroup2",
            "codes": [ "US-NY" ]
        }
    ],
    "queryInfo": {
        "q": "",
        "sort": "name",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 2,
        "offset": 0,
        "returnedCount": 2
    }
}
```

Account-Level IP Directional Group List DTO

This is returned when retrieving multiple Account-Level IP Groups from the server. It is not used for creating or updating Account-Level IP Groups.

Table 42 Account-Level SourceIP Group List DTO Structure

Field	Description	Type
accountName	The account name for the groups.	String.
ipGroups	The list of all Account-Level SourceIP Groups that matched the query for the offset and limit.	List of *Account-Level SourceIP Directional Group DTO*s
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**S**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all events in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: Account-Level SourceIP Group List

```
{
    "accountName": "myAccount",
    "ipGroups": [
        {
            "name": "accountIPGroup",
            "description": "Another sample",
```

```
            "ips": [
                {
                    "start": "1.1.1.1",
                    "end": "2.2.2.2"
                },
                {
                    "address": "4.3.2.1"
                }
            ]
        },
        {
            "name": "accountIPGroup2",
            "ips": [
                {
                    "cidr": "10.10.10.10/30",
                }
            ]
        }
    ],
    "queryInfo" : {
        "q": "",
        "sort": "name",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 2,
        "offset": 0,
        "returnedCount": 2
    }
}
```

CAA Records API

The Certification Authority Authorization, or CAA record, allows for a domain name holder to authorize one or more certification authorities to issue certificates for a domain. Additionally, the records allow for the implementation of additional controls by a Public Certification Authority which can prevent certificate issues. (Certificates are generally valid for at least one year)

The CAA records specify an authorization control to be performed by a certificate issuer before the issuance of a certificate.

A CAA resource record consists of a flags byte and a tag-value pair referred to as a property. Domain names may have multiple CAA RRs (Resource Record that includes the owner name, class, type, time to live, and data) associated with it, so a given property may be specified more than once.

Table 43 CAA DTO Record Types

Field (Tags)	Description	Type
issue	Authorizes the domain name holder to issue certificates for the domain.	Value Types: ▪ **"0"** – Default. ▪ **"1"** (Issuer Critical) - Indicates that the corresponding tag MUST be understood if the record is to be properly interpreted by an issuer. See *Table 44 CAA Record - Additional Tag Values* for acceptable tag formats.
issuewild	Authorizes the domain name holder to issue wildcard certificates for the domain.	Adheres to the same syntax as **issue**, however, will take precedence over **issue** properties if specified.
iodef	Specifies a URL to which an issuer "may" report certificate issue requests that may be inconsistent with the issuer's policies or practices.	HTTP or HTTPS.

The canonical presentation format of the CAA record is:

```
CAA <flags> <tag> <value>
```

Where:

- Flags: Is an unsigned integer between 0 and 255.
- Tag: Is a non-zero sequence of US-ASCII letters and numbers in lower case.

- Value: Is the <character-string> encoding of the value field

The following example depicts a situation in which certificates are not to be issued, except by the holder of the domain name, or an authorized agent.

```
{
   $ORIGIN example.com
      CAA 0 issue "ca.example.net"
}
```

For circumstances in which one more iodef properties are specified, a certificate issuer may report invalid certificate requests to a specific address.

```
{
   $ORIGIN example.com
      CAA 0 issue "ca.example.net"
      CAA 0 iodef "mailto:security@example.com"
      CAA 0 iodef "http://iodef.example.com"
}
```

Certificate issuers may specify additional parameters that allow customers to specify additional parameters in return which can govern the certificate issuance. This could include a Certificate policy number for example to be used or referenced. The following example demonstrates a situation in which domain holder for “ca.example.net” has requested its customers (example.com) to specify an account number in each CAA record.

```
{
   $ORIGIN example.com
      CAA 0 issue "ca.example.net; account=xxxxxx"
}
```

Table 44 CAA Record - Additional Tag Values

Tag Type	Value
issuevalue	= [domain] [“ ; ” *(space parameter)]
domain	= label *(“.” label)
label	= (ALPHA / DIGIT) *(*(“-“) (ALPHA / DIGIT))
space	= *(SP / HTAB)
parameter	= tag “=” value
tag	= 1*(ALPHA / DIGIT)
value	= *VCHAR

Create CAA Records

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/CAA/{zoneName}
```

Parameters: None.

Body: Must include an *RRSet DTO* and *CAA DTO Record Types*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or a {zoneName} you do not have access to.
- If you don't have permission to create CAA Records.
- If a resource record or {zoneName} of the same type already exists.

JSON Example: Create a CAA Record

```
{
   "zoneName": "0-a-accounttest.com.",
   "ownerName": "caarec",
   "rrtype": "CAA",
   "ttl": null,
   "rdata": [
         "3 issue \"a\""
         ]
}
```

Get CAA Records

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/CAA/{zoneName}
```

Parameters: None.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RRSet DTO* and *CAA DTO Record Types* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create CAA Records.

JSON Example: Get a CAA Record

```
{
    "zoneName": "0-a-accounttest.com.",
    "rrSets": [
```

```
        {
            "ownerName": "caarec.0-a-accounttest.com.",
            "rrtype": "CAA (257)",
            "ttl": 86400,
            "rdata": [
                "1 issue \"a a\"",
                "3 issue \"a\""
            ]
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```


Update CAA Records

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/CAA/{zoneName}
```

Parameters: None.

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create CAA Records.

Delete CAA Records

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/CAA/{zoneName}
```

Parameters: None.

Body: None.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create/delete CAA Records.

TLSA Records API

DNS Transport Layer Security Protocol (TLSA)

The Transport Layer Security Protocol, or TLSA provides communication security across the internet, by using channel encryption. The TLSA DNS resource record is used to associate a TLS server certificate or public key with the domain name where the record is found, thereby forming a “TLSA certificate association.”

By using certificates, TLS can bind keys (secret / public) so that they cannot be duplicated or falsified. By combining a published key with additional specific information (i.e. name of a service), and then being signed by a separate key, the TLS records creates in essence, an “anchor” key. When a key is used to validate the signature of certificates being received, it will be validated by the “anchor” key, thereby preventing untrusted signing from occurring.

For example:

"Example.com" can only be signed by the key(s) for "com", and the "com" key(s) can only be signed by the DNS root.

Create TLSA Records

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/TLSA/{zoneName}
```

Parameters: None.

Body: Must include an *RRSet DTO* and *TLSA DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or if it’s not a {zoneName} you have access to.
- If you don't have permission to create TLSA Records.
- If a resource record or {zoneName} of the same type already exists.

JSON Example: Create TLSA Record

```
{
   "ttl": 300,
   "rdata": ["1 0 0 82003ba34942dc74"]
}
```

For a greater detailed explanation of the above JSON example return for Creating TLSA Records, refer to the following table. **Each field and value combination are separated by a space in the api call**.

Table 45 TLSA DTO

Field	Description	Type
Certification Usage	The first integer displayed. Describes the type of Certificate to be used.	Valid values are: ▪ **0** - Specify CA certificate/public key for certificate to certify end server. Certificate might not have the basic Constraints extension present. ▪ **1** - Specify CA certificate/public key for certificate to certify end server. Service certificate constraint. ▪ **2** - Specify CA certificate/public key for certificate to certify end server. Trust anchor assertion. ▪ **3** - Domain-issued certificate (same as 1, with no PKIX validation is tested).
Field-Selector	The second integer displayed. Describes the type of selector to be used.	▪ **0** – Full certificate of the Certificate Binary structure. ▪ **1** – SubjectPublicKeyInfo
Matching Type	The third integer displayed. Describes the type of matching to be used.	▪ **0** – Exact match on selected content. ▪ **1** – SHA-256 hash of selected content. ▪ **2** – SHA-512 hash of selected content.
Certificate-Association-Data	The string of hexadecimal characters that describes certificate associated data.	▪ The certificate-association-data must be represented as a string of hexadecimal characters. Whitespace is allowed within the string of hexadecimal characters.

Get TLSA Records

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/TLSA/{zoneName}
```

Parameters: None.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RRSet DTO* and *TLSA DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create TLSA Records.

JSON Example: Get TLSA Record

```
{
    "zoneName": "0-a-accounttest.com.",
    "rrSets": [
        {
            "ownerName": "_1._tcp.test1.0-a-accounttest.com.",
            "rrtype": "TLSA (52)",
            "ttl": 20,
            "rdata": [
                "0 0 0 12345678",
                "0 0 0 123ba45679"
            ]
        }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```

Update TLSA Records

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/TLSA/{zoneName}
```

Parameters: None.

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create TLSA Records.

Delete TLSA Records

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/TLSA/{zoneName}
```

Parameters: None.

Body: None.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or not a {zoneName} you have access to.
- If you don't have permission to create/delete TLSA Records.

SiteBacker and Traffic Controller Pools API

SiteBacker (SB) and Traffic Controller (TC) pools provide advanced services above what is possible with an RD Pool. Records in a SB or TC pool can have probes attached to them, which verify the functionality of the servers over a variety of protocols. Records that fail to return proper responses within the expected time frame can result in a notification and/or the removal of the records from the pool. If all records in a pool fail to respond to their probes, SB and TC pools can be configured to return backup records instead.

The only valid rrtype value for SiteBacker or Traffic Controller pools is A. While SB and TC pools can intermix CNAME records with A records, it isn't legal to specify a SiteBacker or Traffic Controller pool of type CNAME. Instead, CNAME records can refer to other SB, TC, RD, Directional pools, or can be references to standard RRSets inside or outside of the zone.

Profiles

All pools are implemented as additional information added to *Resource Record Sets* (RRSets). This additional information is specified in a section with the label profile. Every profile contains an entry with the label @context. The value of @context is a URI that uniquely identifies the type of the pool.

The URI for a SiteBacker Pool is "http://schemas.ultradns.com/SBPool.jsonschema".

The URI for a Traffic Controller Pool is "http://schemas.ultradns.com/TCPool.jsonschema".

SiteBacker Profile Fields

The other fields in the profile for a SiteBacker Pool are:

Table 46 SiteBacker Pool Fields

Field	Description	Type
description	An optional description of the SiteBacker pool.	String. Less than 255 characters.
runProbes	Indicates whether or not the probes are run for this pool.	Boolean. If not specified, defaults to true.
actOnProbes	Indicates whether or not pool records will be enabled (**true**) or disabled (**false**) when probes are run.	Boolean. If not specified, defaults to “rue.
order	Indicates the order of the records returned by the resolver for the SiteBacker pool. Valid values are FIXED, RANDOM, and ROUND_ROBIN.	String. If not specified, defaults to ROUND_ROBIN.

Field	Description	Type
maxActive	Specifies the maximum number of active servers in a pool and determines when SiteBacker takes backup servers offline. For example, consider a pool with six servers. Setting Max Active to 4 means SiteBacker takes two servers offline and sends the four active records in the answer.	Integer. Value from 1 to the number of records in the pool. If not specified, defaults to “1.”
failureThreshold	The minimum number of records that must fail for a pool to be labeled 'FAILED'. If the number of failed records in the pool is greater than or equal to the 'Failure Threshold' value, the pool will be labeled FAILED. For example, a pool with six priority records, one all-fail record, and the Failure Threshold value is set to four (4). If four or more priority records are not available to serve, the pool will be labeled FAILED, and the all-fail record will be served.	Long. Valid value between 0 and the number of priority records in the pool. If not specified, defaults to null. **Note**: For value ”0” or ”null,” the failure-threshold logic will be disabled.
maxServed	Determines the number of record answers for each query. This is typically All Active records or a sub set of Max Active.	Integer. Value from 1 to maxActive. If not specified, defaults to maxActive.

Field	Description	Type
status	Returned for all SiteBacker and/or Traffic Controller GET calls. Returns the following results: ▪ **OK** – If the number of records serving is equal to the Max Active value, and all the active records are top priority records. For example, if a pool has a Max Active of 1 and the Priority 1 record is serving. ▪ **WARNING** – If the number of records serving is equal to the Max Active value, and the active records are not top priority records. For example, if a pool has a Max Active of 1 and the Priority 1 records is not serving and the Priority 2 record is serving. ▪ **CRITICAL** – If the number of records serving is less than the Max Active value, or the All Fail record is being served. For example, if a pool has a Max Active of 2, and only one record is serving. ▪ **FAILED** – If the FailureThreshold value is "0" or null, and no records are serving, and there is no All Fail record configured. **OR** If the number of priority records not available to serve equals or exceeds the FailureThreshold's value. (For example, if the Failure Threshold value is 3, and there are 3 or more Priority Records that are not available to serve.)	String. Ignored if present on PUT or PATCH, returned by GET.
rdataInfo	One entry for each entry in rdata. Metadata for each rdata.	Array of rdataInfo structures, in order matching the rdata entries in the main body. See below for rdataInfo structure.
backupRecords	List of backup records for the SiteBacker pool. Specifies the records to be served if all other records fail. There can be one or more A records used as backup records, or a single CNAME record.	Array of backupRecordInfo structures. Optional.
backupRecords/backupRecord	An entry in the list backupRecords list.	backupRecordInfo structure.
backupRecords/backupRecord/rdata	The IPv4 address or CNAME for the backup record.	String. rdata string (for SiteBacker and Traffic Controller, a valid IPv4 address or CNAME).

Field	Description	Type
backupRecords/backupRecord/failoverDelay	Specifies the time, from 0–30 minutes, that SiteBacker waits after detecting that the pool record has failed before activating primary records.	Integer. If not specified, defaults to "0."
backupRecords/backupRecord/availableToServe	Indicates whether the pool is active and available to serve records.	Boolean.

Table 47 RDataInfo Fields

Field	Description	Type
state	The current state of the pool record. Valid values are NORMAL, ACTIVE, and INACTIVE.	String. Defaults to NORMAL.
runProbes	Indicates whether or not probes are run for this pool record.	Boolean. Defaults to true.
priority	Indicates the serving preference for this pool record.	Integer.
failoverDelay	Specifies the time, from 0–30 minutes, that SiteBacker waits after detecting that the pool record has failed before activating secondary records.	Integer. If not specified, defaults to 0 (activate the secondary records immediately).
threshold	Specifies how many probes must agree before the record state is changed.	Integer.
weight	Determines the traffic load to send to each server in the Traffic Controller pool.	Even integers from "2" to "100." If not specified, defaults to "2." Only applies to a record in a Traffic Controller pool. Ignored if present in a POST/PUT/PATCH to a SiteBacker Pool.
availableToServe	Indicates whether the pool is active and available to serve records. Applies to JSON response, not JSON request.	Boolean. Defaults to true.
status	Returned for all SiteBacker and/or Traffic Controller GET calls. Returns the following results: ▪ OK – If the number of records serving is equal to the Max Active value, and all the active records are top priority records. For example, if a pool has a Max Active of 1 and the Priority 1 record is serving. ▪ WARNING – If the number of records serving is equal to the Max Active value, and the active records are not top priority records. For example, if a pool has a Max Active of 1 and the Priority 1 record is not	String.

Field	Description	Type
	serving and the Priority 2 record is serving. ▪ CRITICAL – If the number of records serving is less than the Max Active value, or if the All Fail record is being served. For example, if a pool has a Max Active of 2, and only one record is serving. ▪ FAILED – If no records are serving, and there is no All Fail record configured.	

JSON Example: SiteBacker RRSet with Profile

```
{
    "ttl": 300,
    "rdata": [
        "1.2.3.4",
        "a.domain.name.",
        "9.8.7.6",
        "30.40.50.60"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/SBPool.jsonschema",
        "description": "STRING",
        "runProbes": true,
        "actOnProbes": true,
        "order": "FIXED",
        "maxActive": 1,
        "failureThreshold": 0,
        "maxServed": 1,
        "rdataInfo": [
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 1,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 2,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 3,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            },
```

```
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 4,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            }
        ],
        "backupRecords": [
            {
                "rdata": "8.5.6.7",
                "failoverDelay":5
            },
            {
                "rdata": "9.10.11.12"
            }
        ]
    }
}
```

Traffic Controller Profile Fields

Traffic Controller pools have very similar fields in their profile to those of the Site Backer pools. The differences are:

- The URI @context is "http://schemas.ultradns.com/TCPool.jsonschema".
- maxServed is not allowed (only one record is ever served at a time in a Traffic Controller pool).
- Order is not allowed (again, only one record is ever served at a time).
- maxActive is replaced with maxToLB. The maximum value is the number of pool records. If it is not specified, it defaults to 0.
- backupRecords is replaced by backupRecord, which is a single backupRecordInfo structure. As in SiteBacker, backupRecord is optional.

JSON Example: Traffic Controller RRSet with Profile

```
{
    "ttl": 300,
    "rdata": [
        "1.2.3.4",
        "a.domain.name.",
        "9.8.7.6",
        "30.40.50.60"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/TCPool.jsonschema",
        "description": "STRING",
        "runProbes": true,
        "actOnProbes": true,
        "maxToLB": 3,
        "failureThreshold": 0,
        "rdataInfo": [
```

```
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 1,
                "failoverDelay": 0,
                "threshold": 1,
                "weight": 2,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 2,
                "failoverDelay": 0,
                "threshold": 1,
                "weight": 2,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 3,
                "failoverDelay": 0,
                "threshold": 1,
                "weight": 2,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 4,
                "failoverDelay": 0,
                "threshold": 1,
                "weight": 2,
                "availableToServe": true
            }
        ],
        "backupRecord": {
            "rdata": "9.8.7.6"
        }
    }
}
```


Priorities

The priority feature for SiteBacker and Traffic Controller Pools works by utilizing the integer value that is provided for the *priority* field for each record in a pool. Assigning the value “1” to a record indicates that it is of the highest priority, and should always be returned first. Afterwards, each subsequent priority value will be returned in incremental fashion. Ideally, we request that each record be assigned its own individual priority value so as to avoid conflict with records being returned out of order.

However, should the same priority value be assigned to multiple records within a pool, the following will occur:

- When there are duplicate priorities assigned to records in a pool, the records will first be sorted by the priority integer value, and then will be sorted alphabetically in order using the rdata for the record.

In the following example, our pool contains five records with duplicate priority values.

- **poolrecord2.pool.com** with priority 3
- **1.1.1.1** with priority 3
- **poolrecord1.pool.com** with priority 3
- **poolrecord3.pool.com** with priority 1
- **poolrecrod4.pool.com** with priority 2

JSON Example: SB/TC Priority Ordering

```
{
    "zoneName": "pool.com.",
    "rrSets": [
        {
            "ownerName": "owner.pool.com.",
            "rrtype": "A (1)",
            "ttl": 120,
            "rdata": [
                "poolrecord3.pool.com.",
                "poolrecord4.pool.com.",
                "1.1.1.1",
                "poolrecord1.pool.com.",
                "poolrecord2.pool.com."
            ],
            "profile": {
                "@context": "http://schemas.ultradns.com/TCPool.jsonschema",
                "description": "owner.pool.com.",
                "runProbes": false,
                "actOnProbes": false,
                "status": "OK",
                "rdataInfo": [
                    {
                        "state": "NORMAL",
                        "runProbes": true,
                        "priority": 1,
                        "failoverDelay": 0,
                        "threshold": 1,
                        "weight": 2,
                        "availableToServe": false,
                        "status": "OK"
                    },
                    {
                        "state": "NORMAL",
                        "runProbes": true,
                        "priority": 2,
                        "failoverDelay": 0,
                        "threshold": 1,
                        "weight": 2,
```

```
                    "availableToServe": false,
                    "status": "OK"
                },
                {
                    "state": "ACTIVE",
                    "runProbes": true,
                    "priority": 3,
                    "failoverDelay": 0,
                    "threshold": 1,
                    "weight": 2,
                    "availableToServe": false,
                    "status": "OK"
                },
                {
                    "state": "ACTIVE",
                    "runProbes": true,
                    "priority": 3,
                    "failoverDelay": 0,
                    "threshold": 1,
                    "weight": 2,
                    "availableToServe": false,
                    "status": "OK"
                },
                {
                    "state": "ACTIVE",
                    "runProbes": true,
                    "priority": 3,
                    "failoverDelay": 0,
                    "threshold": 1,
                    "weight": 2,
                    "availableToServe": false,
                    "status": "OK"
                }
            ],
            "maxToLB": 2
        }
    }
    ],
    "queryInfo": {
        "sort": "OWNER",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    }
}
```


Get a Sitebacker or Traffic Controller Pool

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None.

Body: None.

Response: If task completes, Status code 200 OK is returned with *SiteBacker Profile Fields* and/or *Traffic Controller Profile Fields* in the body content.

Errors: An error is returned under the following conditions:

- If this URI does not refer to a SB/TC pool.
- If you do not have permission to access this SB/TC pool.

Get All SiteBacker and Traffic Controller Pools

Method and URI:

To get SiteBacker Pools:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:SB_POOLS
```

To get Traffic Controller Pools:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:TC_POOLS
```

To get both SiteBacker and Traffic Controller Pools together:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:SB_POOLS,
TC_POOLS
```

Parameters: None.

Body: None.

Response: If task completes, Status code 200 OK is returned with *SiteBacker Profile Fields* and/or *Traffic Controller Profile Fields* in the body content.

Errors: An error is returned under the following conditions:

- If this URI does not refer to an SB/TC pool.
- If you do not have permission to access this SB/TC pool.

Users should provide a maximum limit of 1000 when requesting to get all SB/TC pools. If the limit is greater than 1000, the following error message will be returned:

```
{
errorCode: 22000
errorMessage: "Invalid Page Limit, the maximum number of records that can be
retrieved are restricted to 1000."
}
```

Create Sitebacker and Traffic Controller Pools

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with Sitebacker/TrafficController pool profile info, or a *JSON PATCH DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If type is invalid.
- If {ownerName} is not valid.
- If invalid data was submitted in the body around any validation error for Failure Threshold.

Users should provide a FailureThreshold value between zero and "X" when creating SB/TC pools, where the value X is the total number of priority records. If the provided value is less than zero or greater than X, the following error message will be returned:

```
{
"errorCode": 2951,
"errorMessage": "Invalid Failure Threshold. Failure threshold
value should be from 0 to 1."
}
```

Update SiteBacker and Traffic Controller Pools

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with Sitebacker/TrafficController pool profile info, or a *JSON PATCH DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to update records.
- If type is invalid.
- If {ownerName} does not exist.

Users can delete a specific record from a DIR pool by sending a PUT request, without providing that record in the input.

Partially Update SiteBacker and Traffic Controller Pools

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/{rrType}/{ownerName}
```

Parameters: None

Body: Must include an *RRSet DTO* with Sitebacker/TrafficController pool profile info, or a *JSON PATCH DTO*.

Patchable Objects for SiteBacker and Traffic Controller:

- biz.neustar.ultra.rest.dto.SiteBackerPoolProfile
- biz.neustar.ultra.rest.dto.TCPoolProfile

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don't have permission to partial update SB/TC pool.

Delete SiteBacker and Traffic Controller Pools

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{ownerName}
```

Parameters: None

Body: None

Response: If delete happens immediately, Status Code of 204 is returned with no body content.

- If delete happens in the background, a Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to delete records.
- If type is invalid.
- If ownerName does not exist.

Get Probe Alerts

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/alerts
```

Parameters: None.

Body: None.

Response: If task completes, Status Code 200 OK is returned with an *Alert Data List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you don't have permissions to access alerts.
- If this URI does not refer to an SB/TC pool.
- If you do not have permission to access this SB/TC pool.

Alert Data DTO

The API call /alerts returns all probing errors for the current user. This data is returned as AlertData DTOs in an AlertDataList.

Table 48 AlertData DTO Structure

Field	Description	Type
poolRecord	The pool record that triggered the alert.	String.

Field	Description	Type
probeType	The name of the probe type.	String.
probeStatus	The status determined by the probe. Valid values are **OK**, **WARNING**, **CRITICAL**, and **FAILED**.	String.
alertDate	The date the alert occurred.	String, date/time formatted in ISO 8601 format.
failoverOccurred	Flag to indicate if failover occurred.	Boolean.
ownerName	The ownerName of the pool that alerted.	String.
status	Status of the probe. Valid values are Active and Inactive.	String.

Alert Data List DTO

Table 49 AlertData List DTO Structure

Field	Description	Type
alerts	The list of all alerts.	List of *Alert Data DTO*.
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all events in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: AlertDataList

```
{
    "alerts": [
        {
            "status": "Active",
            "ownerName": "host1.example-zone.com.",
            "poolRecord": "172.16.8.1",
            "alertDate": "2014-04-01T08:05:42Z",
            "probeStatus": "FAILED",
            "probeType": "DNS",
            "failoverOccurred": true
        },
        {
            "status": "Active",
            "ownerName": "h.testzone.com.",
            "poolRecord": "1.2.3.4",
            "alertDate": "2014-04-01T17:24:33Z",
            "probeStatus": "FAILED",
```

```
            "probeType": "HTTP",
            "failoverOccurred": true
        }
    ],
    "queryInfo": {
        "q": "",
        "sort": "alertDate",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 2,
        "offset": 0,
        "returnedCount": 2
    }
}
```


TTL Records Consistency in Sitebacker/Traffic Controller Pool Records

Per RFC 2181, the Time To Live (TTL) of all Resource Records in an RRSet must be the same value. Whenever any existing Sitebacker / Traffic Controller Pool record TTL is modified using a partial update (PATCH) request, all other existing Sitebacker / Traffic Controller Pool records will also be updated with the modified TTL value. There will be an audit event for each Sitebacker / Traffic Controller Pool record modification.

Below is an example scenario of the new TTL function for Sitebacker / Traffic Controller pools.

If there is an existing Sitebacker / Traffic Controller pool with two records:

- Record one → Points to 1.1.1.1 with a TTL value of 400
- Record two → Points to 1.2.3.4 with a TTL value of 86400

If the TTL value for Record one (1.1.1.1) is updated using the partial update request to change the value from 400 to 500, then the additional record(s) must be updated as well. In this scenario, Record two (1.2.3.4) will be altered from a TTL value of 86400 to 500.

SiteBacker and Traffic Controller Pool Probes

As of August 27th, 2018, new region names have been added for SiteBacker and Traffic Controller Pool Probe calls. These new region names are not (currently) replacing the previous region names, however, we do recommend you begin to utilize these new region names, as the old region names will eventually not be valid. These new regions provide greater customization and granularity when establishing probe locations to desired regions, or returning more specific probe regions when querying.

Functionally, you can use the old region names along with the new region names without triggering an error. We are supporting backward compatibility through the REST API. This means that you can provide the new region name North_America_East on a POST call, but on a GET call, that same new region name North_America_East is returned as the old region name New_York.

Eventually, the REST API will no longer accep the old region names, nor will it return the old region names as a response. Please begin to use the new region names as we begin to transition these changes through the REST API.

Table 53 below lists the new region names along with the locations that are incorporated with the new region name.

Create a Probe

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes
```

Parameters: None

Body: Must include a *Probe Info DTO*

Response: If task completes, Status Code 201 is returned with a Location Header containing a URI, and the GUID for the created probe in the body content.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.

JSON Example: Create a Sitebacker / Traffic Controller Probe – Pool Level

```
{
            "type": "HTTP",
            "interval": "ONE_MINUTE",
            "agents": [
                "Palo_Alto",
```

```
                "NEW_YORK"
            ],
            "threshold": 2,
            "details": {
                "transactions": [
                    {
                        "method": "GET",
                        "url": "http://www.google1.com/",
                        "transmittedData": "",
                        "limits": {
                            "connect": {
                                "warning": 20,
                                "critical": 20,
                                "fail": 20
                            },
                            "run": {
                                "warning": 60,
                                "critical": 60,
                                "fail": 60
                            }
                        },
                        "followRedirects": true
                    }
                ]
            }
        }
```

Get Probes

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes
```

Parameters

Table 50 Get probes parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are: ▪ **type:** Valid values are RECORD, POOL, or ALL. Default value is ALL, unless poolRecord is specified. ▪ **poolRecord:** If you only want to get probes for a single pool record, specify either the IPv4 or CNAME as a FQDN for the pool record. If specified, type of RECORD is assumed.	String.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Probe Info List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.

- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.

Get a Probe

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes/
{guid}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Probe Info DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the {guid} is not a guid for a probe for the pool or a pool record.

JSON Example: Get a Sitebacker / Traffic Controller Pool Probe via id

```
{
    "id": "060844D2F8B23F19",
    "poolRecord": "1.1.1.1",
    "type": "HTTP",
    "interval": "ONE_MINUTE",
    "agents": [
        "DALLAS",
        "AMSTERDAM",
        "ASIA",
        "NEW_YORK"
    ],
    "threshold": 2,
    "details": {
        "transactions": [
            {
                "method": "GET",
                "url": "http://www.google1.com/",
                "transmittedData": "",
                "limits": {
                    "connect": {
                        "fail": 20
                    },
                    "run": {
```

```
                        "fail": 60
                    }
                },
                "followRedirects": true
            }
        ]
    }
}
```

Get SiteBacker Agents for account

This call allows users to query for supported agents for an account

Table 51 SiteBacker Agent DTO

Field	Description	Type
name	Name of the agent.	String.
location	SB Agent location.	String.
description	SB Agent description.	String.

Table 52 SiteBacker AgentList DTO

Field	Description	Type
agents	List of all the supported SiteBacker Agents.	List of SiteBacker Agents.

Table 53 ProbeType Sitebacker Agent - Updated Region/Agent Names

Field	New Regions	Sites Associated to the Regions
regionName	**ASIA**	Tokyo, Taipei, Singapore, Hong Kong, Sydney
	CHINA	Beijing, Hong Kong *Note: Reliable results only to China based hosts.*
	EUROPE_EAST	Amsterdam, Frankfurt, Stockholm
	EUROPE_WEST	London, Dublin, Paris, Madrid
	NORTH_AMERICA_CENTRAL	Chicago, Denver, Dallas, Minneapolis
	NORTH_AMERICA_EAST	New York, Washington D.C., Atlanta, Miami, Toronto
	NORTH_AMERICA_WEST	Seattle, San Jose, Los Angeles, Phoenix, Vancouver
	SOUTH_AMERICA	Sao Paolo, Colombia, Miami

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/agents
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *SiteBacker AgentList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If an invalid method or URI is provided.

JSON Example: Get Sitebacker Agents


```
{
    "agents": [
        {
            "name": "North America Central",
            "description": "(Chicago, Denver, Dallas)",
            "location": "USCENTRAL"
        },
        {
            "name": "Europe East",
            "description": "(Amsterdam, Frankfurt, Stockholm)",
            "location": "EUEAST"
        },
        {
            "name": "Europe West",
            "description": "(London, Dublin, Paris, Madrid)",
            "location": "EUWEST"
        },
        {
            "name": "South America",
            "description": "(Sao Paulo,Columbia, Miami)",
            "location": "SAMERICA"
        },
        {
            "name": "Asia",
            "description": "(Tokyo, Taipei, Singapore, Hong Kong, Sydney)",
            "location": "ASIA"
        },
        {
            "name": "China",
            "description": "(Beijing, Hong Kong) Note: Reliable results only
                            within China",
            "location": "CHINA"
        },
        {
            "name": "North America West",
            "description": "(Seattle, San Jose, Los Angeles, Phoenix,
                            Vancouver)",
            "location": "USWEST"
        },
        {
            "name": "North America East",
            "description": "(New York, Washington D.C., Atlanta, Miami,
                            Toronto)",
            "location": "USEAST"
        }
    ]
}
```

Update a Probe

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes/{guid}
```

Parameters: None

Body: Must include a *Probe Info DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the {guid} is not a guid for a probe for the pool or a pool record.

Partially Update a Probe

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes/{guid}
```

Parameters: None

Body: Must include a *Probe Info DTO* , *or a JSON PATCH DTO*.

Patchable Objects for SB/TC:

- biz.neustar.ultra.rest.dto.ProbeInfo

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If {ownerName} does not exist.
- If you don't have permission to read this zone.
- If you don't have permissions to access the pool.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.

- If the {guid} is not a defined guid for a probe for the pool or a pool record.

JSON Example: Partially Updating a Probe with new and old regions response

```
{
    "probes": [
        {
            "id": "060844D2F8B23F19",
            "poolRecord": "1.1.1.1",
            "type": "HTTP",
            "interval": "ONE_MINUTE",
            "agents": [
                "DALLAS",
                "AMSTERDAM",
                "ASIA",
                "NEW_YORK"
            ],
            "threshold": 2,
            "details": {
                "transactions": [
                    {
                        "method": "GET",
                        "url": "http://www.google1.com/",
                        "transmittedData": "",
                        "limits": {
                            "connect": {
                                "fail": 20
                            },
                            "run": {
                                "fail": 60
                            }
                        },
                        "followRedirects": true
                    }
                ]
            }
        },
```

Delete a Probe

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/probes/{guid}
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no content in the body.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the {guid} is not a guid for a probe for the pool or a pool record.

Probe Info DTO

Probes are defined in their own endpoints, relative to the pool for pool-level probes, and relative to the records for record-level probes.

Each of the probe types has a different details section. In a ProbeInfo DTO (shown below), the structure of the **details** section must match the structure associated with the value in the type field.

The Probe Info DTO is shown immediately below. The *Probe Details DTOs* for each type are shown in the remaining sections of this chapter.

Table 54 ProbeInfo DTO structure

Field	Description	Type
id	The internal id for this probe. Returned by GET.	String. Always returned by GET. Ignored if present on POST, PUT, or PATCH.
poolRecord	The pool record associated with this probe. ▪ Returned by GET when returning a record-level probe. ▪ Left blank when creating a pool-level probe. ▪ Specified when creating a record-level probe.	String. Sometimes returned by GET, required for POST when creating a record-level probe. Ignored if present on PUT or PATCH.
type	Type of the probe. Valid values are HTTP, PING, FTP, TCP, SMTP, SMTP_SEND, and DNS.	String. Required, no default value.
interval	Length of time between probes in minutes. Value values are HALF_MINUTE, ONE_MINUTE, TWO_MINUTES, FIVE_MINUTES, TEN_MINUTES, and FIFTEEN_MINUTES.	String. If not specified, defaults to FIVE_MINUTES.
agents	Locations that will be used for probing. One or more values must be specified. Valid values are NEW_YORK, PALO_ALTO, DALLAS, and AMSTERDAM. See *Table 53 ProbeType Sitebacker Agent - Updated Region/Agent Names* for the new agent names.	Array of Strings. Required, no default value.
threshold	Number of agents that must agree for a probe state to be changed.	Integer, from 1 to the number of agents specified. Required.
details	Probe type-specific information.	Map of the type-specific fields for a probe. See below.

Probe Info List DTO

This is used to return a list of Probes for a Pool or a Pool Record.

Table 55 ProbeInfo List DTO Structure

Field	Description	Type
probes	The list of all probes.	List of *Probe Info DTO*s
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all events in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: Probe Info List with two TCP Probes

```
{
   "probes": [
      {
         "type": "TCP",
         "interval": 5,
         "agents": [
            "NEW_YORK",
            "DALLAS"
         ],
         "threshold": 1,
         "details": {
            "port": 1024,
            "controlIP": "1.2.3.4",
            "limits": {
               "connect": {
                  "warning": 20,
                  "critical": 20,
                  "fail": 20
               },
               "avgConnect": {
                  "warning": 20,
                  "critical": 20,
                  "fail": 20
               },
            }
         }
      },
      {
         "type": "TCP",
         "interval": 5,
```

```
        "agents": [
           "NEW_YORK",
           "DALLAS"
        ],
        "threshold": 1,
        "details": {
           "port": 2048,
           "controlIP": "1.2.3.4",
           "limits": {
              "connect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "avgConnect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
           }
        }
     }
  ],
  "queryInfo": {
        "q":"",
        "sort": "type",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 2,
        "offset": 0,
        "returnedCount": 2
    }
}
```


Probe Details DTOs

The below DTOs provide the data and structure needed for the **details** field of the *Probe Info DTO*. The Details DTOs you will use is determined by the type of the probe identified.

HTTP Probe Details DTO

Table 56 HTTP Probe Details DTO structure

Field	Description	Type
transactions	List of http requests sent for a single probe.	Array of transaction info structures.
transactions/method	HTTP method. Valid values are GET or POST.	String. Required.
transactions/url	URL to probe.	String. Required.

Field	Description	Type
transactions/transmittedData	Data to send to URL.	String. Optional.
transactions/followRedirects	Indicates whether or not to follow redirects.	Boolean. Optional, defaults to false.

transactions/expectedResponse	The Expected Response code for probes to be returned as Successful. Valid values are:
	• 2XX: Probe will pass for any code between 200-299.
	• 3XX: Probe will pass for any code between 300-399.
	• 2XX|3XX: Probe will pass for any code between 200-399.
	• Any combination of HTTP codes between 100-599 separated by ' | ' For example:
	  ○ 201|302
	  ○ 301|202|401
	  ○ 501|201|404|301
		String.
		Optional.
		By default, probes will be passed for 2XX HTTP code.

transactions/limits	Determine the cutoffs for sending notification or failing the probe.	httpLimitInfo structure.
transactions/limits/connect		
transactions/limits/connect/warning	How long the probe stays connected to the resource to trigger a warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/connect/critical	How long the probe stays connected to the resource to trigger a critical warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/connect/fail	How long the probe stays connected to the resource to trigger a fail (maximum 20 seconds).	Integer. Optional.
transactions/limits/avgConnect		
transactions/limits/avgConnect/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
transactions/limits/avgConnect/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/avgConnect/fail	Mean connect time over the five most recent probes run on each agent to trigger a fail.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/run		
transactions/limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/run/fail	How long the probe should run to trigger a fail.	Integer. Optional.
transactions/limits/avgRun		
transactions/limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a fail.	Integer. Optional, only used for Traffic Controller Pools.
transactions/limits/searchString		

Field	Description	Type
transactions/limits/searchString/warning	If left blank, the HTTP probe verifies that the server responds to the request with a successful HTTP response (normally a Status Code 200). If specified, the probe searches the response's body only; it does not search the status line and headers. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional, only used for Traffic Controller Pools.
transactions/limits/searchString/critical	If left blank, the HTTP probe verifies that the server responds to the request with a successful HTTP response (normally a Status Code 200). If specified, the probe searches the response's body only; it does not search the status line and headers. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional, only used for Traffic Controller Pools.
transactions/limits/searchString/fail	If left blank, the HTTP probe verifies that the server responds to the request with a successful HTTP response (normally a Status Code 200). If specified, the probe searches the response's body only; it does not search the status line and headers. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional.
totalLimits	The total amount of time spent on all http transactions.	httpTotalLimit structure.
totalLimits/warning	Run time for all steps in the sequence of an HTTP transactional probe for a warning to be generated.	Integer. Optional, only used for Traffic Controller Pools.
totalLimits/critical	Run time for all steps in the sequence of an HTTP transactional probe for a critical warning to be generated.	Integer. Optional, only used for Traffic Controller Pools.
totalLimits/fail	Run time for all steps in the sequence of an HTTP transactional probe for the probe to fail.	Integer. Optional.

Ping Probe Details DTO

Table 57 Ping Probe Details DTO structure

Field	Description	Type
packets	Number of ICMP packets to send.	Integer, defaults to 3.
packetSize	Size of packets in bytes.	Integer, defaults to 56.
limits	Determine the cutoffs for sending a notification or failing the probe.	pingLimitInfo structure.
limits/lossPercent		
limits/lossPercent/warning	Percentage of packets lost to trigger a SiteBacker/Traffic Controller warning event. For example, 5 would indicate that packet loss > 5% would trigger a probe failure. 0 always fails the probe; 100 ensures the probe succeeds.	Integer. Optional, only used for Traffic Controller Pools.
limits/lossPercent/critical	Percentage of packets lost to trigger a SiteBacker/Traffic Controller critical warning event. For example, 5 would indicate that packet loss > 5% would trigger a probe failure. 0 always fails the probe; 100 ensures the probe succeeds.	Integer. Optional, only used for Traffic Controller Pools.
limits/lossPercent/fail	Percentage of packets lost to trigger a SiteBacker/Traffic Controller failure event. For example, 5 would indicate that packet loss > 5% would trigger a probe failure. 0 always fails the probe; 100 ensures the probe succeeds.	Integer. Optional.
limits/total		
limits/total/warning	Run time for all pings to complete for a warning to be generated.	Integer. Optional, only used for Traffic Controller Pools.
limits/total/critical	Run time for all pings to complete for a critical warning to be generated.	Integer. Optional, only used for Traffic Controller Pools.
limits/total/fail	Run time for all pings to complete for a failure to be generated.	Integer. Optional
limits/average		
limits/average/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer, optional, only used for Traffic Controller Pools.
limits/average/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/average/fail	Mean connect time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.
limits/run		
limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/fail	How long the probe should run to trigger a failure.	Integer. Optional.
limits/avgRun		
limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.

FTP Probe Details DTO

Table 58 FTP Probe Details DTO structure

Field	Description	Type
port	Which Port to connect to.	Integer between 1 and 65535. Defaults to 21.
passiveMode	Whether or not to use FTP Passive mode.	Boolean, defaults to false.
username	Username for FTP service.	String. Optional.
password	Password for FTP service.	String. Optional.
path	Path to check for a file.	String.
limits	Determine the cutoffs for sending a notification or failing the probe.	ftpLimitInfo structure.
limits/connect		
limits/connect/warning	How long the probe stays connected to the resource to trigger a warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/critical	How long the probe stays connected to the resource to trigger a critical warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/connect/fail	How long the probe stays connected to the resource to trigger a failure (maximum 20 seconds).	Integer. Optional.
limits/avgConnect		
limits/avgConnect/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/fail	Mean connect time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.
limits/run		
limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/fail	How long the probe should run to trigger a failure.	Integer. Optional.
limits/avgRun		
limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.
limits/searchString		
limits/searchString/warning	If blank, the FTP probe verifies that the server responds to the request with a successful response. If specified, the probe searches the response's body. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/searchString/critical	If blank, the FTP probe verifies that the server responds to the request with a successful response. If specified, the probe searches the response's body. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional, only used for Traffic Controller Pools.
limits/searchString/fail	If blank, the FTP probe verifies that the server responds to the request with a successful response. If specified, the probe searches the response's body. If the probe does not find the string within the response, the probe attempts to match it as a regular expression.	String. Optional.

TCP Probe Details DTO

Table 59 TCP Probe Details DTO structure

Field	Description	Type
port	Which Port to connect to.	Integer between 1 and 65535. Required.
controlIP	Provides a control mechanism that allows the web administrators to stop the TCP port on the control system.	String, IP address. Optional.
limits		
limits/connect		
limits/connect/warning	How long the probe stays connected to the resource to trigger a warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/critical	How long the probe stays connected to the resource to trigger a critical warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/fail	How long the probe stays connected to the resource to trigger a failure (maximum 20 seconds).	Integer. Optional.
limits/avgConnect		
limits/avgConnect/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/avgConnect/fail	Mean connect time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.

SMTP Probe Details DTO

Table 60 SMTP Probe Details DTO structure

Field	Description	Type
port	The Port that will be connected to.	Integer between 1 and 65535. Defaults to 25.
limits		
limits/connect		
limits/connect/warning	How long the probe stays connected to the resource to trigger a warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/critical	How long the probe stays connected to the resource to trigger a critical warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/fail	How long the probe stays connected to the resource to trigger a failure (maximum 20 seconds).	Integer. Optional.
limits/avgConnect		
limits/avgConnect/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/fail	Mean connect time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.
limits/run		
limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/fail	How long the probe should run to trigger a fail.	Integer. Optional.
limits/avgRun		
limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.

SMTP_SEND Probe Details DTO

Table 61 SMTP Send Probe Details DTO structure

Field	Description	Type
port	The Port that will be connected to.	Integer between 1 and 65535. Defaults to 25.
from	Email address that will send the message.	String. Email address. Required.
to	Email address that will receive the message.	String. Email address. Required.
message	Email message body.	String. Optional.
limits		
limits/connect		
limits/connect/warning	How long the probe stays connected to the resource to trigger a warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/critical	How long the probe stays connected to the resource to trigger a critical warning (maximum 20 seconds).	Integer. Optional, only used for Traffic Controller Pools.
limits/connect/fail	How long the probe stays connected to the resource to trigger a failure (maximum 20 seconds).	Integer. Optional.
limits/avgConnect		
limits/avgConnect/warning	Mean connect time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/critical	Mean connect time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgConnect/fail	Mean connect time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.
limits/run		
limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/run/fail	How long the probe should run to trigger a fail.	Integer. Optional.
limits/avgRun		
limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.

DNS Probe Details DTO

Table 62 DNS Probe Details DTO structure

Field	Description	Type
port	The Port that should be used for DNS lookup.	Integer between 1 and 65535. Defaults to 53.
tcpOnly	Indicates whether or not the probe should use TCP only, or first UDP then TCP.	Boolean. Defaults to false.
type	Select which kind of record should be checked for. Valid values are NULL, AXFR, or any Resource Record Type.	String. Defaults to NULL.
ownerName	Selects the name that should be queried.	String. Defaults to blank.
limits		
limits/run		
limits/run/warning	How long the probe should run to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/critical	How long the probe should run to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/run/fail	How long the probe should run to trigger a fail.	Integer. Optional.
limits/avgRun		
limits/avgRun/warning	Mean run time over the five most recent probes run on each agent to trigger a warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/critical	Mean run time over the five most recent probes run on each agent to trigger a critical warning.	Integer. Optional, only used for Traffic Controller Pools.
limits/avgRun/fail	Mean run time over the five most recent probes run on each agent to trigger a failure.	Integer. Optional, only used for Traffic Controller Pools.

Field	Description	Type
limits/response		
limits/response/warning	Match exactly, records with single field responses (that is: A, CNAME, DNAME, NS, MB, MD, MF, MG, MR, PTR) to trigger a warning. Match HINFO records partially and without considering case. Match partially for types with multiple field responses, and join all fields separated by spaces (for example, use 10 mail.example.com. to test a response to an MX record with a preference of 10 and target host of mail.example.com.).	String. Optional, only used for Traffic Controller Pools.
limits/response/critical	Match exactly, records with single field responses (that is: A, CNAME, DNAME, NS, MB, MD, MF, MG, MR, PTR) to trigger a critical warning. Match HINFO records partially and without considering case. Match partially for types with multiple field responses, and join all fields separated by spaces (for example, use 10 mail.example.com. to test a response to an MX record with a preference of 10 and target host of mail.example.com.).	String. Optional, only used for Traffic Controller Pools.
limits/response/fail	Match exactly, records with single field responses (that is: A, CNAME, DNAME, NS, MB, MD, MF, MG, MR, PTR) to trigger a failure. Match HINFO records partially and without considering case. Match partially for types with multiple field responses, and join all fields separated by spaces (for example, use 10 mail.example.com. to test a response to an MX record with a preference of 10 and target host of mail.example.com.).	String. Optional.

JSON Example: HTTP Probe Info

```
{
   "type": "HTTP",
   "poolRecord": "1.1.1.1",
   "interval": "ONE_MINUTE",
   "agents": [
      "NEW_YORK",
      "DALLAS"
   ],
   "threshold": 1,
   "details": {
      "transactions": [
```

```
        {
           "method": "POST",
           "url": "https://www.cnn.com/",
           "transmittedData": "foo=bar",
           "followRedirects": true,
           "expectedResponse": "2XX",
           "limits": {
              "connect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "avgConnect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "run": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "avgRun": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "searchString": {
                 "warning": "missing",
                 "critical": "uh-oh",
                 "fail": "bad"
              }
           }
        },
        {
           "method": "GET",
           "url": "https://www.bing.com/",
           "followRedirects": false,
           "expectedResponse":"201|301",
           "limits": {
              "connect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },
              "avgConnect": {
                 "warning": 20,
                 "critical": 20,
                 "fail": 20
              },

              "run": {
                 "critical": 20,
                 "fail": 20
              },
              "avgRun": {
```

```
                  "warning": 20,
                  "critical": 20,
                  "fail": 20
               },
               "searchString": {
                  "warning": "missing",
                  "critical": "uh-oh",
                  "fail": "bad"
               }
            }
         }
      ],
      "totalLimits": {
         "warning": 20,
         "critical": 20,
         "fail": 20
      }
   }
}
```


Sitebacker Agent / Probes

Please make the necessary modifications to your firewall security policies to include the following ninety (90) IP addresses below. These are the IP addresses that will be used by Prober for Traffic Management (Sitebacker) pools.

Table 63 Active IP Probes by Region

Active IP Probes by Region			
North America - East	**North America - West**	**North America - Central**	**Europe - East**
4.28.137.86	**4.31.99.86**	**4.34.39.214**	**212.72.53.214**
4.28.137.87	**4.31.99.87**	**165.254.103.86**	**212.72.53.215**
4.31.108.214	**4.34.39.215**	**165.254.103.87**	**212.73.224.150**
4.31.108.215	**4.34.119.23**	**165.254.103.150**	**212.73.224.151**
4.34.39.229	**4.53.108.215**		**213.130.49.214**
4.34.119.22	**165.254.23.150**		**213.130.49.215**
165.254.102.86	**165.254.23.151**		**213.198.94.215**
165.254.102.87	**165.254.103.22**		
165.254.103.151	**165.254.103.23**		
165.254.103.215	**209.153.240.88**		

Active IP Probes by Region			
Europe - West	**South America**	**Asia**	**China**
83.231.151.150 209.173.59.22 212.187.140.150	156.154.86.22 165.254.103.214 200.15.1.150	61.58.41.22 61.120.158.150 116.51.28.214 202.68.78.214 203.131.248.22	203.131.248.36 204.74.96.22

The following Probes and Regions are going to be available during 2019. Please plan to add the following IP Probes to your Whitelist if you have not already done so.

Table 64 IP Probes Expansion by Region

IP Probes by Region Available 2019		
Region	**IPv4**	**IPv6**
North America – East	156.154.35.153 156.154.35.154 156.154.37.153 156.154.37.154 156.154.119.153 156.154.119.154	2610:a1:3008:128::153 2610:a1:3008:128::154 2610:a1:3010:128::153 2610:a1:3010:128::154 2610:a1:3044:128::153 2610:a1:3044:128::154
North America – West	156.154.36.153 156.154.36.154 156.154.38.153 156.154.38.154 156.154.41.153 156.154.41.154	2610:a1:300c:128::153 2610:a1:300c:128::154 2610:a1:3014:128::153 2610:a1:3014:128::154 2610:a1:3020:128::153 2610:a1:3020:128::154
North America – Central	156.154.39.153 156.154.39.154 156.154.40.153 156.154.40.154	2610:a1:301c:128::153 2610:a1:301c:128::154 2610:a1:3018:128::153 2610:a1:3018:128::154

IP Probes by Region Available 2019		
Region	**IPv4**	**IPv6**
Europe - East	**156.154.76.153** **156.154.76.154** **156.154.77.153** **156.154.77.154** **156.154.78.153** **156.154.78.154** **156.154.80.153** **156.154.80.154**	**2610:a1:302c:128::153** **2610:a1:302c:128::154** **2610:a1:3028:128::153** **2610:a1:3028:128::154** **2610:a1:3038:128::153** **2610:a1:3030:128::153** **2610:a1:3030:128::154** **2610:a1:3038:128::154**
Europe – West	**156.154.79.153** **156.154.79.154** **156.154.85.153** **156.154.85.154**	**2610:a1:3034:128::153** **2610:a1:3034:128::154** **2610:a1:3040:128::153** **2610:a1:3040:128::154**
South America	**156.154.122.153** **156.154.122.154** **156.154.123.153** **156.154.123.154**	**2610:a1:304c:128::153** **2610:a1:304c:128::154** **2610:a1:3048:128::153** **2610:a1:3048:128::154**
Asia	**156.154.99.153** **156.154.99.154** **156.154.180.153** **156.154.180.154** **156.154.181.153** **156.154.181.154** **156.154.182.153** **156.154.182.154** **203.119.15.153** **203.119.15.154**	**2610:a1:305c:128::153** **2610:a1:305c:128::154** **2610:a1:3054:128::153** **2610:a1:3054:128::154** **2610:a1:3058:128::153** **2610:a1:3058:128::154** **2610:a1:3068:128::153** **2610:a1:3068:128::154** **2610:a1:3070:128::153** **2610:a1:3070:128::154**

IP Probes by Region Available 2019		
Region	**IPv4**	**IPv6**
China	**156.154.62.153** **156.154.62.154** **156.154.63.153** **156.154.63.154**	**2610:a1:3074:128::153** **2610:a1:3074:128::154** **2610:a1:3078:128::153** **2610:a1:3078:128::154**

The comprehensive list of IPv4 Addresses is as follows:

- 4.28.137.86
- 4.28.137.87
- 4.31.99.86
- 4.31.99.87
- 4.31.108.214
- 4.31.108.215
- 4.34.39.214
- 4.34.39.215
- 4.34.39.229
- 4.34.119.22
- 4.34.119.23
- 4.53.108.215
- 61.58.41.22
- 61.120.158.150
- 83.231.151.150
- 116.51.28.214
- 156.154.35.153
- 156.154.35.154
- 156.154.36.153
- 156.154.36.154
- 156.154.37.153
- 156.154.37.154
- 156.154.38.153
- 156.154.38.154
- 156.154.62.153
- 156.154.62.154
- 156.154.63.153
- 156.154.63.154
- 156.154.76.153
- 156.154.76.154
- 156.154.77.153
- 156.154.77.154
- 156.154.78.153
- 156.154.78.154
- 156.154.79.153
- 156.154.79.154
- 156.154.80.153
- 156.154.80.154
- 156.154.85.153
- 156.154.85.154
- 156.154.86.22
- 156.154.99.153
- 156.154.99.154
- 156.154.119.153
- 156.154.119.154
- 156.154.122.153
- 156.154.122.154
- 156.154.123.153
- 156.154.182.154
- 165.254.23.150
- 165.254.23.151
- 165.254.102.86
- 165.254.102.87
- 165.254.103.22
- 165.254.103.23
- 165.254.103.86
- 165.254.103.87
- 165.254.103.150
- 165.254.103.151
- 165.254.103.214
- 165.254.103.215
- 200.15.1.150
- 202.68.78.214
- 203.119.15.153
- 203.119.15.154
- 203.131.248.22
- 203.131.248.36
- 204.74.96.22
- 209.153.240.88
- 209.173.56.22
- 212.72.53.214
- 212.72.53.215

- 156.154.39.153
- 156.154.39.154
- 156.154.40.153
- 156.154.40.154
- 156.154.41.153
- 156.154.41.154
- 156.154.123.154
- 156.154.180.153
- 156.154.180.154
- 156.154.181.153
- 156.154.181.154
- 156.154.182.153
- 212.73.224.150
- 212.73.224.151
- 212.187.140.150
- 213.130.49.214
- 213.130.49.215
- 213.198.94.215

The comprehensive list of IPv6 address is as follows:

- 2610:a1:300c:128::153
- 2610:a1:300c:128::154
- 2610:a1:301c:128::153
- 2610:a1:301c:128::154
- 2610:a1:302c:128::153
- 2610:a1:302c:128::154
- 2610:a1:304c:128::153
- 2610:a1:304c:128::154
- 2610:a1:305c:128::153
- 2610:a1:305c:128::154
- 2610:a1:3008:128::153
- 2610:a1:3008:128::154
- 2610:a1:3010:128::153
- 2610:a1:3010:128::154
- 2610:a1:3014:128::153
- 2610:a1:3014:128::154
- 2610:a1:3018:128::153
- 2610:a1:3018:128::154
- 2610:a1:3020:128::153
- 2610:a1:3020:128::154
- 2610:a1:3028:128::153
- 2610:a1:3028:128::154
- 2610:a1:3030:128::153
- 2610:a1:3030:128::154
- 2610:a1:3034:128::153
- 2610:a1:3034:128::154
- 2610:a1:3038:128::153
- 2610:a1:3038:128::154
- 2610:a1:3040:128::153
- 2610:a1:3040:128::154
- 2610:a1:3044:128::153
- 2610:a1:3044:128::154
- 2610:a1:3048:128::153
- 2610:a1:3048:128::154
- 2610:a1:3054:128::153
- 2610:a1:3054:128::154
- 2610:a1:3058:128::153
- 2610:a1:3058:128::154
- 2610:a1:3068:128::153
- 2610:a1:3068:128::154
- 2610:a1:3070:128::153
- 2610:a1:3070:128::154
- 2610:a1:3074:128::153
- 2610:a1:3074:128::154
- 2610:a1:3078:128::153
- 2610:a1:3078:128::154

SiteBacker and Traffic Controller Pool Events

Create an Event

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events
```

Parameters: None

Body: Must include an *EventInfo DTO*.

Response: If task completes, Status Code 201 is returned with a Location Header containing a URI, and the GUID for the created event in the body content.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the poolRecord is not a pool record in the pool.

JSON Example: Create a SB / TC Event

```
{
    "poolRecord": "1.2.3.4",
    "type": "NORMAL",
    "start": "2018-10-02T12:00:00Z",
    "repeat": "WEEKLY",
    "end": "2020-12-31T23:59:59Z",
    "notify": "ALWAYS"
}
```

Get Events for a Pool

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events
```

Parameters:

Table 65 Get events parameters

Parameter	Description	Type
q	Specify query operators to filter the returned result. Valid operators are: ▪ poolRecord - The CNAME or IPv4 for the pool record. This will only return events for the specified pool record.	String.
offset	The position in the list for the first returned element (0 based). The default value is 0.	Integer.
limit	The maximum number of rows requested. The default value is 100.	Integer.
sort	The sort column used to order the list. Sort columns are: ▪ START – the start date for the event. ▪ END – the end date for the event. Note that if you get all events in a pool, they will be primarily sorted by pool record. This sort is a secondary sort.	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). The default value is false.	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *EventInfoList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the poolRecord is not a pool record in the pool.

Get a Single Event

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events/{guid}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *EventInfo DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.

- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the poolRecord is not a pool record in the pool.
- If the {guid} is not a guid for an event for the pool record.

JSON Example: Get a SB / TC Event Response

```
{
    "queryInfo": {
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    },
    "events": [
        {
            "id": "0608452906095FDE",
            "poolRecord": "1.2.3.4",
            "type": "ACTIVE",
            "start": "2018-10-02T12:00:00Z",
            "end": "2020-12-31T23:59:00Z",
            "repeat": "WEEKLY",
            "notify": "ALWAYS"
        }
    ]
}
```

Update an Event

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events/{guid}
```

Parameters: None

Body: Must include an *EventInfo DTO.*

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.

- If you don't have permissions to access the pool.
- If the {guid} is not a guid for an event in the pool.

Partially Update an Event

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events/{guid}
```

Parameters: None

Body: Must include an *EventInfo DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If task happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If the {guid} is not a guid for an event in the pool.

Delete an Event

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/events/{guid}
```

Parameters: None

Body: None

Response: If delete completes immediately, Status Code 204 with no content in the body is returned.

- If delete happens in the background, Status Code 202 is returned along with an X-Task-ID header and status message of Pending in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.

- If the {guid} is not a guid for an event in the pool.

Pool Event DTOs

EventInfo DTO

The EventInfo DTO contains the data to create, get, or update a pool event.

Table 66 EventInfo DTO Structure

Field	Description	Type
id	The internal id for this event. Returned by GET.	String. Always returned by GET, ignored if present on POST, PUT, or PATCH.
poolRecord	The pool record associated with this event. This must be a FQDN if the pool record is a CNAME or reference to another pool, or an IPv4.	String. Required for POST, ignored if present on PUT or PATCH, returned by GET.
type	Indicates what will happen when the event is triggered. Valid values are: **NORMAL** - treat as normal pool record. **ACTIVE** - force the record to be active. **ACTIVE_TEST** - force the record to be active and run a probe test, but do not act on the result. **INACTIVE** - force the record to be inactive. **INACTIVE_TEST** - force the record to inactive and run a probe test, but do not act on the result.	String. One of the valid values. Required.
start	Start date and time for the event.	String. Date/time in ISO 8601 format. Must be in the future.
repeat	How frequently events are triggered. Valid values are DAILY, WEEKLY, FORTNIGHTLY, MONTHLY.	String. If not specified, event does not repeat.
end	The date to stop triggering events.	String. Date in ISO 8601 format after the start date/time. Only allowed if repeat is specified.
notify	Indicates when to notify after an event is triggered. Valid values are: **NEVER** - do not send a notification. **ERROR** - only send notification on error. **ALWAYS** - send notification on success or error.	String. If not specified, defaults to ERROR.

EventInfoList DTO

This is used to return the list of all events for a pool record.

Table 67 EventInfo List DTO

Field	Description	Type
events	The list of returned events. Each entry in the list matches the EventInfo DTO described above.	List of EventInfo DTO.
queryInfo/q	The query used to construct the list.	String.

Field	Description	Type
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all events in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

SiteBacker and Traffic Controller Pool Notifications

Notifications are used to specify who is alerted when an event is triggered for a particular pool record. Since events can only be created for pool records, notifications in turn can only be created for pool records, not for sub pools however.

Notifications can only be created for Primary Records. You cannot create notifications for All Fail Records. If the request contains an All Fail Record, then an error is returned with the following message - **Notifications cannot be configured for All Fail Records.**

Get Notifications for a Pool

The get Notifications API will return notifications for Primary Records only. If any notification was previously created on All Fail record then it will not be returned in the response.

Method and URI

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/notifications
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *NotificationList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.

Create a Notification for Specified Pool Records

Method and URI:

```
POST
https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/notifications/
{emailAddress}
```

Parameters: None

Body: Must include a *Notification DTO*.

Response: If task completes, Status Code 201 is returned with a Location Header containing the URI of the notification.

- If the task happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If {emailAddress} is already associated with any pool records in this pool.
- If the request contains an All Fail Record.

JSON Example: SB pool with 2 Primary Records and one Backup(All Fail) Record

```
{
    "ttl": 300,
    "rdata": [
        "1.2.3.4",
        "a.domain.name."
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/SBPool.jsonschema",
        "description": "STRING",
        "runProbes": true,
        "actOnProbes": true,
        "order": "FIXED",
        "maxActive": 1,
        "failureThreshold": 0,
        "maxServed": 1,
        "rdataInfo": [
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 1,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            },
            {
                "state": "NORMAL",
                "runProbes": true,
                "priority": 2,
                "failoverDelay": 0,
                "threshold": 1,
                "availableToServe": true
            }
        ],
        "backupRecords": [
            {
                "rdata": "9.10.11.12"
            }
        ]
    }
}
```

JSON Example: Create a Pool Notification for Primary Records

```
{
    "email": "email@notification.com",
    "poolRecords": [
        {
            "poolRecord": "1.2.3.4",
            "notification": {
                "probe": true,
                "record": false,
                "scheduled": true
            }
        },
        {
            "poolRecord": "a.domain.name.",
            "notification": {
                "probe": true,
                "record": true,
                "scheduled": false
            }
        }
    ]
}
```

JSON Example: Create a Pool Notification that contains an All Fail Record

```
{
    "email": "email@notification.com",
    "poolRecords": [
        {
            "poolRecord": "1.2.3.4",
            "notification": {
                "probe": true,
                "record": false,
                "scheduled": true
            }
        },
        {
            "poolRecord": "9.10.11.12",
            "notification": {
                "probe": true,
                "record": true,
                "scheduled": false
            }
        }

     ]
}
```

This request will throw an error – ‘Notifications cannot be configured for All Fail Records’.

Get a Notification for a Pool

Method and URI

```
GET
https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/notifications/
{emailAddress}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *NotificationList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If this {emailAddress} has not been defined for notifications for this pool.

Update a Notification

This will update the Notification Information for only the Pool Record specified in the *Notification DTO*. If the notify flag (probe, scheduled, record) is not specified in the Notification DTO, then it is set to false.

If you try to update a notification for an All Fail Record, an error with the following message will occur - **Notifications cannot be configured for All Fail Records**

The following sample JSON will turn off Notification for a Pool Record:

```
{
      "email": "test@test1.com",
      "poolRecords": [
                {
                  "poolRecord": "1.2.3.4",
                  "notification": {
                    }
                }
            ]
}
```

Method and URI

```
PUT
https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/notifications/
{emailAddress}
```

Parameters: None

Body: Must include a *Notification DTO*. Because this is a full update, if the notification flag is not specified in the DTO, the notification will be set to false (turned off). See the JSON example above.

Response: If update completes, Status Code 200 OK is returned with an appropriate status message in the response body.

- If update happens in the background, Status Code 202 is returned along with an X-Task-ID header and a status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.
- If you don't have permissions to access the pool.
- If {emailAddress} is not associated with any pool records in this pool.
- If the request contains an All Fail Record.
- If the input contains duplicate records.

Delete a Notification

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/A/{ownerName}/
notifications/{emailAddress}
```

Parameters: None

Body: None

Response: If delete completes immediately, Status Code 204 is returned with no content in the body.

- If delete happens in the background, Status Code Status Code 202 is returned along with an X-Task-ID header and status message of “Pending” in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.
- If pool does not exist or is not a SiteBacker/Traffic Controller pool.

Pool Notification DTOs

NotificationInfo DTO

The NotificationInfo DTO describes the conditions in which a notification is sent.

Table 68 NotificationInfo DTO

Field	Description	Type
probe	Indicates whether to notify on probe events.	Boolean.
record	Indicates whether to notify on record events.	Boolean.
scheduled	Indicates whether to notify on scheduled events.	Boolean.

JSON Example: Notification

```
{
   "probe": true,
   "record": true,
   "scheduled": false
 }
```

Notification DTO

The Notification DTO associates an email address with a list of pool records. Each pool record is in turn associated with a NotificationInfo.

Table 69 Notification DTO

Field	Description	Type
email	The email address being notified. This field is returned on GET, but ignored on POST, PUT, and PATCH.	String. Valid email address.
notifications/poolRecords	The list of pool record settings for this email address.	Array.
notifications/poolRecords /poolRecord	The pool record.	String. Either an IP address or a valid owner name.
notifications/poolRecords /notification	The NotificationInfo DTO describing the notifications for this pool record/email address combination.	*NotificationInfo DTO*.

JSON Example: Notification Info

```
{
   "email": "a@b.com",
   "poolRecords": [
      {
         "poolRecord": "1.2.3.4",
         "notification": {
            "probe": true,
            "record": false,
            "scheduled": true
```

```
            }
         },
         {
            "poolRecord": "2.4.5.6",
            "notification": {
               "probe": true,
               "record": true,
               "scheduled”: false
            }
         }
      ]
   }
```

NotificationList DTO

The NotificationList encapsulates the configured notifications.

Table 70 NotificationList DTO

Field	Description	Type
notifications	The list of notifications.	Array of the Notification DTO.

JSON Example: Notification Info List

```
{
   "notifications": [
      {
         "email": "a@b.com",
         "poolRecords": [
            {
               "poolRecord": "1.2.3.4",
               "notification": {
                  "probe": true,
                  "record": false,
                  "scheduled": true
               }
            },
            {
               "poolRecord": "2.4.5.6",
               "notification": {
                  "probe": true,
                  "record": true,
                  "scheduled": false
               }
            }
         ]
      },
      {
         "email": "c@d.com",
         "poolRecords": [
            {
               "poolRecord": "1.2.3.4",
               "notification": {
                  "probe": false,
                  "record": true,
```

```
                    "scheduled": false
                }
            }
        ]
    }
  ]
}
```

Simple Monitor / Failover Pool API

Simple Failover (SF) Pools are used to define a single address (the live record), a simple HTTP monitor, and a backup address. If the monitor detects that the live record is unreachable from too many global regions, the backup (Failover) record is shown. The user can also force the backup record to be live, or unforce it.

Simple Monitoring is a feature that is a subset of Simple Failover and Simple Load Balancing (SLB). You can run all the Simple Failover calls as a Simple Monitor call instead by ensuring that your Failover record information is removed from the body of the call. Examples will be provided for each call.

Simple Monitoring (SM) is designed to provide single resource record sites with a very basic website availability monitor. This monitor tracks if a website is available or unreachable, and alerts the customer to unavailability via email notification. This feature does not provide fail over assistance to an alternative record (i.e. All fail), nor does it provide any measurement statistics on the health of the website (beyond whether the site is available or down).

The most common example of a Simple Monitor would be a customer having only one IP Address, and needs a monitor to notify them when the address has an outage, instead of needing to failover to another address.

Simple Monitor/ Failover pools now support the usage of IPv6 addresses by using the "AAAA" record type. You will need to alter the API calls to match the AAAA record type when trying to use an IPv6 instead of an IPv4 address that uses the A record type.

IPv6 Address Format	Example
Ipv6 Full Notation	"2001:0db8:0a0b:12f0:0000:0000:0000:0001"
Compressed	"2001:db8:a0b:12f0::1" **OR** "2610:a1:1059:0:0:0:0:35"

Modifications of Existing DTOs and Parameters

Pools are implemented as additional information added to RRSets. A SF Pool DTO has an optional section called "Profile". The Profile contains information that is outside the bounds of the various DNS specifications.

Profile Information for Simple Monitor / Failover Pools

A profile is simply a JSON Map. All profiles must contain a key called @context. This is a JSON-LD reference to a schema (http://json-ld.org/). The schema describes the custom data that is used in the profile fields.

The schema name for Simple Monitor / Failover pools is:
http://schemas.ultradns.com/SFPool.jsonschema.

The other fields in the profile for a Simple Monitor / Failover pool are:

Table 71 Simple Failover Pool Profile Fields

Field	Description	Type
poolDescription	An optional description of the Simple Failover field.	String. Less than 255 characters. Optional.
liveRecordDescription	An optional description of the live record.	String. Less than 255 characters. Optional.
liveRecordState	Whether or not the live record is currently active. This field is optional for create or update, but is not returned for a list of RRSETS.	One of either: FORCED_INACTIVE or NOT_FORCED. ▪ FORCED_INACTIVE – the backup record should always be returned by a DNS query. ▪ NOT_FORCED – the monitor should determine if the live record or the backup record is returned by a DNS query.
backupRecord	Information for the backup record.	A backupRecord (All Fail) DTO. Required for Simple Failover pools. Ignored for Simple Monitor pools.
backupRecord/rdata	BIND data for the backup record.	IPv4 address or IPv6 address. Required for Simple Failover pools. Ignored for Simple Monitor pools.
backupRecord/description	An optional description for the backup record.	String. Less than 255 characters. Optional for Simple Failover pools. Ignored for Simple Monitor pools.
monitor	Information for the monitor.	A monitor DTO. Required.
monitor/method	HTTP method used to connect to the monitored URL.	One of either GET or POST. Required.
monitor/url	Monitored URL.	A full URL including: protocol, host, and URI. Required. Valid protocols are http and https.
monitor/transmittedData	If a monitor is sending a POST, the data that is sent as the body of the request.	String. Less than 255 characters. Optional.
monitor/searchString	If supplied, a string that is checked against the returned data from the request. The monitor fails if the searchString is not present.	String. Less than 255 characters. Optional.
regionFailureSensitivity	Specifies the sensitivity to the number of regions that need to fail for the backup record to be made active.	Required. One either LOW or HIGH: ▪ LOW – All 4 regions have to fail. ▪ HIGH – 2 or more regions have to fail.

Field	Description	Type
status	Ignored if sent on create or update. Returned when list of RRSets is returned.	One of: OK, CRITICAL, MANUAL ▪ OK – Live record is being served. ▪ CRITICAL – The backup record is being served due to the monitor detecting a failure. ▪ MANUAL – The backup record is being served due to user forcing the live record to be inactive.

JSON Example: Simple Monitor / Failover Pool Profile

```
{
   "zoneName": "domain.name.",
   "ownername": "a.domain.name.",
   "rrtype": "A",
   "ttl": 300,
   "rdata": [
       "1.2.3.4",
   ],
   "profile": {
       "@context": "http://schema.ultradns.com/SFPool.jsonschema",
       "pooldescription": "Pool description",
       "liveRecordDescription": "Live Record Description",
       "liveRecordState": "FORCED_INACTIVE",
       "backupRecord": {
           "rdata": "5.6.7.8.",
           "description": "backup record description",
      },
   "monitor": {
       "method": "POST",
       "url": "http://www.cnn.com/",
       "transmittedData": "foo=10&bar=20",
       "searchString": "testing",
      }
   "regionFailureSensitivity": "LOW",
   "status": "OK",
   }
}
```

- Simple Monitor / Failover Pools can only be defined for RRSets of type A (The DNS value of an A record is displayed as a value of 1. For the complete list of DNS values, they can be found here: https://en.wikipedia.org/wiki/List_of_DNS_record_types). It is an error to define a Simple Failover Pool for other RRSet types.
- It is an error to define a Simple Monitor / Failover Pool with any number of records other than one.

Create Simple Monitor / Failover Pools

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{newPoolName}
```

Parameters: None

Body: Must include *Simple Failover Pool Profile Fields*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.

JSON Example: Create Simple Monitor / Failover Pool

```
{
   "zoneName": "domain.name.",
   "ownerName": "a.domain.name.",
   "rrtype": "A",
   "ttl": 300,
   "rdata": [
      "1.2.3.4"
      ]
   "profile": {
      "@context": "http://schema.ultradns.com/SFPool.jsonschema",
      "poolDescription": "Pool description",
      "liveRecordDescription": "Live Record description",
      "liveRecordState": "FORCED_INACTIVE",
      "backupRecord": {
         "rdata": "5.6.7.8",
         "description": backup record description"
         },
   "monitor": {
      "method": "POST",
      "url": "http://www.cnn.com/",
      "transmittedData": "foo=10&bar=20",
      "searchString": "testing"
         },
      "regionFailureSensitivity": "LOW"
      }
}
```

Simple Monitor and Simple Failover pools use the same API calls, but you will see a slight difference in the "**liverecordstate**". A Simple Monitor pool will display the following: `"liveRecordState": "NOT_FORCED" or "NULL".`

List Simple Monitor / Failover Pools

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/{type}
```

OR

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:SF_POOLS
```

Parameters: Include one of the following:

Table 72 List Configuration Fields

Value	Meaning
ALL	All pools and records (same as RECORDS, POOLS)
RECORDS	Only resource records.
POOLS	All pools.
RD_POOLS	Only RD pools.
DIR_POOLS	Only Directional pools.
SF_POOLS	Only Simple Monitor / Failover pools.
SLB_POOLS	Only Simple Load Balancing Pools

The q parameter recognizes an attribute, **kind**. The valid values for kind are:

- These values can be comma-separated.
- If **kind** is not specified, it defaults to the value ALL.

Body: None.

Response: If task completes, Status Code 200 OK is returned with *Simple Failover Pool Profile Fields* in the body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.

JSON Example: List Simple Monitor / Failover Pools Sample Response

```
{
   "zoneName": "primary-example.",
   "rrsets": [
   {
      "ownerName": "sf.primary-exmaple.com",
      "rrtype": "A(1)",
      "ttl": 300
      "rdata": [
         "1.2.3.4"
         ]

   "profile": {
      "@context": "http://schema.ultradns.com/SFPool.jsonschema",
      "poolDescription": "Pool description",
```

```
      "liveRecordDescription": "Live Record description",

      "backupRecord": {
         "description": "backup record description",
         "rdata": "5.6.7.8"
         },
   "monitor": {
      "method": "POST",
      "url": "http://www.cnn.com/",
      "transmittedData": "foo=10&bar=20",
      "searchString": "testing"
         },
      "regionFailureSensitivity": "LOW"
      "status": "MANUAL",
      }
   }
]
   "queryInfo": {
      "sort": "OWNER",
      "reverse": false,
      "limit": " 100
      }
   "resultInfo": {
      "totalCount": 1,
      "offset": 0,
      "returnedCount": 1
   }
}
```

Simple Monitor and Simple Failover pools use the same API calls, but you will see a slight difference in the "**liverecordstate**". A Simple Monitor pool will display the following: `"liveRecordState": "NOT_FORCED" or "NULL".`

Update Simple Monitor / Failover Pools

For full updates (PUT), the Simple Monitor / Failover Pool profile must be fully specified.

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{newPoolName}
```

Parameters: None

Body: Must include *Simple Failover Pool Profile Fields*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.

JSON Example: Update for Simple Monitor / Failover Pools

```
{
   "zoneName": "domain.name",
   "rrSets": [
    {
     "ownername": "a.domain.name",
     "rrtype": "A (1)",
     "ttl": 300,
     "rdata": [
       "4.3.2.1"
   ]
   "profile": {
      "@context": "http://schema.ultradns.com/SFPool.jsonschema",
      "poolDescription": "Pool description",
      "liveRecordDescription": "Live Record description",
      "backupRecord": {
         "description": "backup record description",
         "rdata": "8.7.6.5",
      }
   "monitor": {
      "method": "POST",
      "url": "http://www.cnn.com",
      "transmittedData": "foo=10&bar=20",
      "searchString": "testing"
      }
   "regionFailureSensitivity": "LOW"
   "status": "MANUAL"
   }
}
```

Partially Update Simple Monitor / Failover Pools

For partial updates (PATCH) that do not affect the sorting order or pool / live record, the profile section is not required.

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{newPoolName}
```

Parameters: None

Body: Must include *Simple Failover Pool Profile Fields*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid.
- If you don't have permission to read this zone.

JSON Example: Partial Update Simple Monitor / Failover Pools Sample Request Body

```
{
   "rdata": [
      "1.1.1.1"
   ]
   "profile": {
      "backupRecord": {
         "rdata": "2.2.2.2"
         "description": "backup record description",
      }
   }
}
```

Conversions for Traffic Service Pools

Conversions to and from Simple Monitor / Failover pools are allowed only via the PUT operation. Conversions via PATCH API are not currently supported for these pool types.

Converting existing A records to Simple Monitor / Failover Pools

To convert an existing owner of a single A record to a Simple Monitor / Failover pool, perform a full update (PUT) and include the profile information for a Simple Monitor / Failover Pool.

Converting a Simple Monitor / Failover Pool to an A record

To convert a Simple Monitor / Failover Pool to an owner of a single A record, perform a full update (PUT) and do not include the profile information. There can only be a single rdata record specified.

Delete Simple Monitor / Failover Pools

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/
{ExistingPoolName}
```

Parameters: None.

Body: None.

Response: If delete happens immediately, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- If {zoneName} is not valid, or you do not have access to it.
- If you don't have permission to delete the {zoneName}.

The {ExistingPoolName} can be either a Fully Qualified Domain Name, or a Relative Domain Name.

Simple Load Balancing (SLB) Pool API

Simple Load Balancing Pools are used to define a pool of up to five (5) IPv4 A records (Primary Records pool), an HTTP(S) monitor, and a backup (All Fail) IPv4 address. One resource record will be served based on the Response Method configured.

Simple Load Balancing supports the usage of IPv6 addresses, using the AAAA record type. You will need to alter the API calls to match the AAAA record type when trying to use an IPv6 address instead of an IPv4 address.

An additional limitation is that you are not able to mix and match IPv4 and IPv6 addresses in the rdata field of the body of a call. You must keep these record types consistent throughout the call.

The following is the list of available IPv6 formats that are accepted:

IPv6 Address Format	Example
IPv6 Full Notation	“2001:0db8:0a0b:12f0:0000:0000:0000:0001”
Compressed	“2001:db8:a0b:12f0::1” "2610:a1:1059:0:0:0:0:35"

When using Simple Load Balancing, the defined monitor (probe) sends HTTP(S) GET or POST requests from four locations towards the target addresses once every five minutes. Optionally, the request to the target system can include HTTP(S) request data and/or the HTTP response data can be searched for specific content. If no search string is specified, the probe of the target is considered as successful if any non-error HTTP response from the target is received. The availability of the target system is evaluated upon receipt of each successful or unsuccessful probe result from each location.

If the monitor (probe) detects that a record in the Primary Records pool is unreachable, that record will not be served unless the record's Forced State is set to Forced Active. If the record's Forced State is set to Forced Inactive, a record will not be served regardless of the pool probe status.

Serving Preference determines if records will be selected from the **Primary Records** pool or from the **All Fail Record**.

1. *Auto Select* (default): Serving method switches from serving **Primary Records**, to **All Fail Record** based upon probe results, and the **Forced State** of the **Primary Records**.
2. *Serve Primary*: Only the **Primary Records** are served based upon the probe results and the **Forced State** of the **Primary Records**.
3. *Serve All Fail*: Only the **All Fail Record** will be served, ignoring the probe results and the **Forced State** of the **Primary Records**.

Note:

- *Auto Select* and *Serve Primary* will also take into account the **Response Method** that has been established (if applicable) when determining which records get served.

- It may take up to 15 seconds to see the updated value after a switching between *Auto Select*/*Serve Primary* and *Serve All Fail*.

Calling the APIs

In this document you will find descriptions of the Data Transfer Objects (DTOs) and the various available services. In all cases, you will receive a 2xx HTTP response code in response to a successful service call. If an error condition occurs, you will receive an HTTP response code in the 4xx or 5xx range, with a HTTP body that indicates the error condition such as:

```
<errors>
      <code>AccountNotFound<code>
      <message>Account not found with ID: abc123xyz</message>
</errors>
```

The UltraDNS APIs can accept requests and send responses in both XML and JSON formats. The client can control the format of the request and the response, by supplying the "Content-Type" and "Accept" HTTP headers respectively, as well as supplying either an "**application/xml**" or "**application/json**" for the value in either header. The default response format is JSON.

Modifications of Existing DTOs and Parameters

Pools are implemented as additional information added to RRSets. A SLBPool DTO has a section called “profile.” The profile contains information that is outside the typical boundaries of various DNS specifications.

Profile Information for Simple Load Balancing Pools

A profile is simply a JSON Map. All profiles must contain a key called @context. This is a JSON-LD reference to a schema (http://json-ld.org/). The schema describes the custom data that is used in the profile fields.

The schema name for SLB Pools is set to http://schemas.ultradns.com/SLBPool.jsonschema.

The other fields in the profile for a SLB Pool are:

Table 73 SLB Pool Profile Information

Field	Description	Type
description	An optional description of the Simple Load Balancing field.	String. Less than 255 characters. Optional.
rdataInfo	The data used to describe the pool records.	Array of *RData Info DTO*. **Required**.
responseMethod	The method used to select which record is returned from the primary record pool.	Required. One of the following: ▪ **PRIORITY_HUNT** – The sequence of the records listed in the primary record pool determines the priority. The first record listed is the highest priority record. Once a record starts being served, it will always be served until the probe detects a

Field	Description	Type
		failure on this record or the record is FORCED_INACTIVE. **The top priority record is always returned among all the set of primary records where the following conditions are satisfied:** 1) Pool Probe is determined to be passing successfully for the record (based upon Threshold configuration), and the record forced state is NOT_FORCED and probingEnabled at this record level is set to true. 2) Record forcedState is set to FORCED_ACTIVE. ▪ **RANDOM – A random record is returned from the following set of primary records.** 1) Pool Probe is determined to be passing successfully (based upon Threshold configuration), and the record forced state is NOT_FORCED and probingEnabled at this record level is set to true. 2) Record forcedState is set to FORCED_ACTIVE. ▪ **ROUND_ROBIN - A record is returned in rotation from the following set of records.** 1) Pool Probe is determined to be passing successfully (based upon Threshold configuration), and the record forced state is NOT_FORCED and probingEnabled at this record level is set to true. 2) If record forcedState is set to FORCED_ACTIVE.
allFailRecord	Information for the backup record.	An *All Fail Record DTO*. Required.
monitor	Information for the probe / monitor.	A *Monitor DTO*. Required.
regionFailureSensitivity	Specifies the sensitivity to the number of regions that must fail probing (Low or High) in order for the backup record to become active.	Required. One of either: ▪ **LOW** – All 4 regions have to fail. ▪ **HIGH** – 2 or more regions have to fail.

Field	Description	Type
servingPreference	Allows users to specify which set of records to serve, from the Simple Load Balancing Pool. Possible choices are: **AUTO_SELECT, SERVE_PRIMARY, SERVE_ALL_FAIL.**	Required. One of : ▪ **AUTO_SELECT** – The pool can switch from serving from the primary record set, to an All Fail record OR, from an All Fail record to the primary record set automatically, based upon the probe results and other record forcedState settings. ▪ **SERVE_PRIMARY** – Only a record in the primary pool will be served. There is no fail over to the “All fail” record. ▪ **SERVE_ALL_FAIL** – Only the All Fail Record will be served..
status	Ignored if sent on Create or Update. Returned when list of RRSets is returned.	One of either: ▪ **OK-** Priority record(s) are being served. ▪ **WARNING** – One of the priority records is not being served due to the monitor detecting a failure, but there is still a priority record to be served. ▪ **CRITICAL** – The backup All Fail record is being served due to the monitor detecting a failure. This is a pool level setting that is determined based upon the number of records available to be served. The number of records available to be served is determined by the probe results from various regions.

RData Info DTO

Table 74 RData Info DTO

Field	Description	Type
rdataInfo/description	An optional description of the record in the live pool.	String. Less than 255 characters. Optional.
rdataInfo/forcedState	The Forced State of the record that indicates whether the	Optional. Valid values are one of the following:

Field	Description	Type
	record needs to be: force served, forced to be inactive, or the force status not being set all. **Defaults to NOT_FORCED**.	▪ **FORCED_ACTIVE** ▪ **FORCED_INACTIVE** ▪ **NOT_FORCED** (default)
rdataInfo/probingEnabled	Can be set at the record level to indicate whether probing is required or not for the given record. **Defaults to true at the record level.**	true or false.
rdataInfo/availableToServe	Indicates whether the record is available to be served (true) or not (false), based upon the probe results or the forced state of the record. Applies to JSON response, not JSON request.	true or false.

All Fail Record DTO

Table 75 All Fail Record DTO

Field	Description	Type
allFailRecord/rdata	BIND data for the backup record.	IPv4 or IPV6 address. Required.
allFailRecord/description	An optional description for the backup record.	String. Less than 255 characters. Optional.
allFailRecord/serving	Serving status of the AllFail Record. This is a response only attribute. true = serving false = not serving	true or false.

Monitor DTO

Table 76 Monitor DTO

Field	Description	Type
monitor/method	The HTTP method used to connect to the monitored URL.	Required. One of either: ▪ **GET** ▪ **POST** ▪ **PUT**
monitor/url	The monitored URL.	Required. A full URL including protocol, host, and URI. Valid protocols are http and https.
monitor/transmittedData	If a monitor is sending a POST, the data that is sent as the body of the request.	String. Less than 255 characters. Optional.
monitor/searchString	If supplied, the string that is checked against the returned data from the request. The monitor fails if the search string is not present.	String. Less than 255 characters. Optional.

Create a Simple Load Balancing Pool

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{newPoolName}
```

Parameters: None

Body: Must include *Profile Information for Simple Load Balancing Pools*.

Response: If task completes, Status Code 201 is returned with a *Task DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If you do not have permission to read {taskId}.

The {newPoolName} can either be a Fully Qualified Domain Name, or a Relative Domain Name.

The below JSON example demonstrates creating a new pool along with five (5) records, along with the All Fail record.

JSON Example: Create SLB Pool IPv4 Address

```
{
    "ttl": 333,
    "rdata": [
        "1.1.1.1",
        "2.2.2.2",
        "3.3.3.3",
        "4.4.4.4",
        "5.5.5.5"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/SLBPool.jsonschema",
        "description": "SLB Pool description",
        "rdataInfo": [
            {
                "description": "1",
                "forcedState": "NOT_FORCED",
                "probingEnabled": true
            },
            {
                "description": "2",
                "forcedState": "FORCED_ACTIVE",
                "probingEnabled": false
            },
            {
                "description": "3",
                "forcedState": "FORCED_INACTIVE",
                "probingEnabled": true
            },
            {
                "description": "4",
                "forcedState": "NOT_FORCED",
                "probingEnabled": true
            },
            {
                "description": "5",
                "forcedState": "NOT_FORCED",
                "probingEnabled": true
            }
        ],
        "responseMethod": "PRIORITY_HUNT",
        "allFailRecord": {
            "rdata": "6.6.6.6",
            "description": "Backup Record"
        },
        "monitor": {
            "method": "POST",
            "url": "https://www.google.com",
            "transmittedData": "q=something",
            "searchString": "UltraDNS"
        },
        "regionFailureSensitivity": "LOW",
        "servingPreference": "AUTO_SELECT"
    }
}
```

- Simple Load Balancing Pools (SLB) can only be defined for RRSets of type A (1). It is an error to define a SLB Pool for other RRSets.
- It is also an error to define a SLB pool with no records, or with more than five (5) records.

List Simple Load Balancing Pools

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/rrsets/?q=kind:SLB_POOLS
```

Parameters:

The “q” parameter for GET /v1/zones/{zoneName}/rrsets and GET /zones/{zoneName}/rrsets/{type} recognize a new attribute called “kind.” The valid values for kind are:

Value	Meaning
ALL	All pools and records (such as RECORDS, and POOLS).
RECORDS	Only resource records.
POOLS	All Pools.
RD_POOLS	Only RD Pools.
DIR_POOLS	Only Directional Pools.
SF_POOLS	Only Simple Monitor / Failover Pools.
SLB_POOLS	Only Simple Load Balancing Pools.

- These values can be comma-separated.
- If **kind** is not specified, it will default to the value **ALL**.
- This list is subject to change as new pool types are defined.

Body: None

Response: If task completes, Status Code 200 OK is returned with an *RData Info DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have access to the zoneName.
- If the zoneName is invalid.

How Simple Load Balancing Pools are Displayed

When an owner that represents a SLB Pool is returned, the profile information must be included.

If SLB Pools are included in a list with other RRSet types, the SLB Pools are placed at the end of the list. If the list is paginated, the SLB Pools are returned after the pages with all other record types listed.

The hierarchy for the display order is as follows:

1) Standard RRSets, RD Pools, and DIR Pools intermixed.
2) All SF Pools
3) All SLB Pools.

JSON Example: Displaying SLB Pools IPv4

```
STATUS 200 OK
{
    "zoneName": "primary-example.com.",
    "rrSets": [
            {
            "ownerName": "primary-example.com.",
            "rrtype": "A (1)",
            "ttl": 333,
            "rdata": [
                 "1.1.1.1",
                 "2.2.2.2",
                 "3.3.3.3",
                 "4.4.4.4",
                 "5.5.5.5"
                    ],
            "profile": {
                "@context": "http://schemas.ultradns.com/SLBPool.jsonschema",
                "description": "SLB Pool description",
                "rdataInfo": [
                     {
                       "description": "1",
                       "forcedState": "NOT_FORCED",
                       "probingEnabled": true,
                       "availableToServe": true
                     },
                     {
                       "description": "2",
                       "forcedState": "FORCED_ACTIVE",
                       "probingEnabled": false,
                       "availableToServe": true
                     },
                     {
                       "description": "3",
                       "forcedState": "FORCED_INACTIVE",
                       "probingEnabled": true
                       "availableToServe": false,
                     },
                     {
                       "description": "4",
                       "forcedState": "NOT_FORCED",
                       "probingEnabled": true,
                       "availableToServe": true
```

```
                        },
                        {
                         "description": "5",
                         "forcedState": "NOT_FORCED",
                         "probingEnabled": true,
                         "availableToServe": true
                         }
                         ],
                "responseMethod": "PRIORITY_HUNT",
                "allFailRecord": {
                                  "description": "Backup Record"
                                  "serving": false
                                  "rdata": "6.6.6.6",

                                   },
                "monitor": {
                          "method": "POST",
                          "url": "https://www.google.com",
                          "transmittedData": "q=something",
                          "searchString": "UltraDNS"
                              },
                "regionFailureSensitivity": "LOW",
                "status":"OK"
                "servingPreference": "AUTO_SELECT",

                }
            }
        ]
}
```

Update Simple Load Balancing Pools

For full updates (PUT), the SLB Pool profile must be fully specified.

Modify both the Pool Configuration and Records / Backup Record

Method and URI:

```
PUT
https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{existingPoolName}
```

- The {existingPoolName} can either be a Fully Qualified Domain Name, or a Relative Domain Name.

When performing a PUT request on a Simple Load Balancing pool, if the Transmitted Data is being completed as a GET call, you will need to remove the information inside the " " of the Transmitted Data section. Otherwise, you will enounter an error.

```
"monitor": {
            "method": "PUT",
            "url": "http://www.google.com",
            "transmittedData": " ",
            "searchString": "UltraDNS"
}
```

JSON Example: Full Update IPv4

```
{
    "ttl": 333,
    "rdata": [
        "1.1.1.1",
        "2.2.2.2",
        "3.3.3.3",
        "5.5.5.5"
    ],
    "profile": {
        "@context": "http://schemas.ultradns.com/SLBPool.jsonschema",
        "description": "SLB Pool description",
        "rdataInfo": [
            {
                "description": "1"
                "forcedState": "NOT_FORCED"
                "probingEnabled": true
            },
            {
                "description": "2"
                "forcedState": "FORCED_ACTIVE"
                "probingEnabled": false
            },
            {
                "description": "3"
                "forcedState": "FORCED_INACTIVE"
                "probingEnabled": true
            },
            {
                "description": "5"
                "forcedState": "NOT_FORCED"
                "probingEnabled": true
            }
        ],
        "responseMethod": "RANDOM",
        "allFailRecord": {
            "rdata": "7.7.7.7",
            "description": "Backup Record"
        },
        "monitor": {
            "method": "POST",
            "url": "http://www.google.com",
            "transmittedData": "q=something",
            "searchString": "UltraDNS"
        },
        "regionFailureSensitivity": "HIGH"
        "servingPreference": "SERVE_PRIMARY"
    }
}
```

Adding a Record into a Simple Load Balancing Pool

The Adding-Record (AR) operation is a special case of the general SLB Partial Update Pool steps outlined in this guide, and it follows the same URL and content format as shown below.

Method and URI:

```
PATCH
https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{existingPoolName}
```

- The {existingPoolName} can either be a Fully Qualified Domain Name, or a Relative Domain Name.

There are two ways to perform this action: Either use a body of *RRSet DTO* or use a body of the *JSON PATCH DTO.*

- If using the RRSet DTO method, the body contains the fields that need to be partially updated, which are structurally similar to what is used in Creating a Pool.

JSON Example: Update Pool via JSON Patch

```
{
  "rdata": ["1.1.1.2"],
  "profile": {
  "rdataInfo": [
        {
            "description": "desc",
            "forcedState": "NOT_FORCED"
        }
    ]
 }
```

- If using the JSON Patch DTO, the following two methods need to be done in the same PATCH location.

JSON Example: Partial Update Record via JSON Patch

(Content Type: application/json-patch+json)

```
[
    {"op": "add", "path": "/rdata/-", "value": "1.1.1.2"},
    {"op": "add", "path": "/profile/rdataInfo/-", "value": {
            "description": "desc",
            "probingEnabled": true
         "forcedState": "Not_Forced"
        }
    }
]
```

- Both of the above examples achieve the same result on an existing SLB pool.

Remove a Record in an Simple Load Balancer Pool

JSON Example: Remove a Record in an SLB Pool

```
[
    {"op": "remove", "path": "/rdata/0"},
    {"op": "remove", "path": "/profile/rdataInfo/0"}
]
```

Updating One or More Configurations in an SLB Pool

JSON Example: Update Pool Level Configurations

```
[
    {"op": "replace", "path": "/profile/description", "value":
      "Pool Level Desc B"}
    {"op": "replace", "path": "/profile/allFailRecord/rdata", "value":
      "1.1.1.1"},
    {"op": "replace", "path": "/profile/responseMethod", "value":
      "ROUND_ROBIN"},
    {"op": "replace", "path": "/profile/monitor/url", "value":
      "http://www.twitter.com"}
    {"op": "replace", "path": "/profile/regionFailureSensitivity",
      "value": "LOW"}
    {"op": "replace", "path": "/profile/servingPreference", "value":
      "SERVE_ALL_FAIL"}
]
```

Delete a Simple Load Balancing Pool

Method and URI:

`DELETE`
`https://api.ultradns.com/zones/{zoneName}/rrsets/{type}/{existingPoolName}`

Parameters: None

Body: None

Response: If delete happens immediately, Status Code of 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If {zoneName} is not a valid ZoneName.

The {existingPoolName} can either be a Fully Qualified Domain Name, or a Relative Domain Name.

The delete endpoints are not affected by the addition of SLB pools, and will remain the same as the other types of pools.

Test Probe

Overview

The Test Probe API allows you to test a probe before setting up the pools that will be associated with a probe. The Test Probe is independent of pool type.

Test Probe supports the use of IPv6 Addresses. Unlike Simple Load Balancing Pools, Test Probes can be a combination of IPv4 and IPv6 addresses when listed in the body of the call.

Create a Test Probe

Method and URI:

```
POST https://api.ultradns.com/testprobes
```

Parameters: None

Body: Must include a *Test Probe DTO*.

Response: If task completes, Status Code 200 OK is returned with *Test Probe Results* in the body content.

Errors: An error is returned under the following conditions:

JSON Example: Test Probe Request

```
{
    "hosts": [
        "1.1.1.1",
        "2.2.2.2",
        "3.3.3.3",
        "4.4.4.4"
    ],
    "type": "HTTP"
    "method": "POST",
    "url": "https://www.google.com",
    "transmittedData": "q=something",
    "searchString": "UltraDNS"
}
```

Test Probe DTO

Table 77 Test Probe DTO

Parameter	Description	Type
hosts	IP addresses of the hosts (probe targets) to be tested.	Array of IP addresses. Required.
type	Test Probe type. Currently limited to HTTP probing type.	Required. Values include: ▪ HTTP

Parameter	Description	Type
		▪ HTTPS
method	Test Probe method. Valid values for HTTP probing are: ▪ GET ▪ POST	String. Required.
url	URL to the probe.	String. Required.
transmittedData	Data that will be sent to the URL.	String. Optional.
searchString	The string to be searched in the HTTP response that comes back from the Probe Target.	String. Optional.
followRedirect	A Boolean flag used to enable (true) /disable (false) the auto HTTP redirection for test probe.	Boolean. Optional. Default value of false.

Test Probe Results

Table 78 Test Probe Results

Parameter	Description	Type
probeResults	The results of probing for each record. Applicable to a response JSON.	Array of Test Probe Results.
probeResults/host	The host that the probe results correspond to. Always present in a response JSON.	IP address.
probeResults/status	Indicator to show if the Probe failed or succeeded for the given record. Values are: ▪ 0 = Success ▪ 3 = Failure If a search string is not specified, the probe is considered to be a Success if any non-error response from the target is received. If a search string is provided, then the success response will be checked for the presence of the string. Always present in a response JSON.	String.
probeResults/message	Message string that accompanies the status of the probeResults. Always present in a response JSON.	String.

JSON Example: Test Probe Response

```
{
    "probeResults": [{
            "host": "1.1.1.1"
            "status": "3",
            "message": "Could not find 'UltraDNS' in the response"
        },
        {
            "host": "2.2.2.2"
            "status": "0"
            "message": "Success"
```

```
        },
        {
            "host": "3.3.3.3"
            "status": "3",
            "message": "404 Not Found"
        },
        {
            "host": "4.4.4.4"
            "status": "0"
            "message": "Success"
        }
```

JSON Example: Test Probe Request with Follow Redirect flag as false

```
{
    "hosts": ["google.com"],
    "type": "HTTP"
    "method": "GET",
    "url": "http://google.com",
    "transmittedData": "q=something",
    "searchString": "UltraDNS",
    "followRedirect": false
}
```

JSON Example: Test Probe Response with Follow Redirect flag as false

```
{
    "probeResults": [
        {
            "host": "google.com",
            "status": "3",
            "message": "301 Moved Permanently to http://www.google.com/"
        }
    ]
}
```

JSON Example: Test Probe Request with Follow Redirect flag as true

```
{
    "hosts": ["google.com"],
    "type": "HTTP"
    "method": "GET",
    "url": "http://google.com",
    "transmittedData": "q=something",
    "searchString": "UltraDNS",
    "followRedirect": true
}
```

JSON Example: Test Probe Response with Follow Redirect flag as true

```
{
    "probeResults": [
        {
            "host": "google.com",
            "status": "0",
            "message": "Success"
        }
```

```
        ]
}
```

Tasks

The Task API allows a user to discover the state of jobs that run in the background. Certain commands that will take longer to complete will return a Task Id in the X-Task-Id header. Use this value to query the state of the task, retrieve data associated with the task, and to remove information about the completed task from the system.

Task DTOs

Task DTO

This describes the current state of a task.

Table 79 Task DTO

Field	Description	Type
taskId	Id for the task.	UUID.
code	Current state of the task.	Use one from: PENDING, IN_PROCESS, COMPLETE, ERROR.
message	Current message for the task.	String.
resultUri	If task is COMPLETE, the URI from where the resulting data can be downloaded.	URI.

JSON Example: Task Status

```
{
    "taskId": "0b40c7dd-748d-4c49-8506-26f0c7d2ea9c",
    "Code": "COMPLETE",
    "message": "Processing complete",
    "resultUri": "/tasks/0b40c7dd-748d-4c49-8506-26f0c7d2ea9c/result"
}
```

TaskList DTO

The TaskList is returned when requesting the state of all tasks for a user.

Table 80 TaskList DTO

Field	Description	Type
tasks	The list of returned tasks. Each entry in the list matches the task DTO described above.	Task DTO.
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.

Field	Description	Type
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: Tasks

```
{
   "tasks": [
      {
         "taskId":"0b40c7dd-748d-4c49-8506-26f0c7d2ea9c",
         "Code":"COMPLETE",
         "message":"Processing complete",
         "resultUri":"/tasks/0b40c7dd-748d-4c49-8506-26f0c7d2ea9c/result"
      }
   ],
   "queryInfo": {
      "q": "",
      "sort": "CODE",
      "reverse": false,
      "limit": 100
   },
   "resultInfo": {
      "totalCount": 1,
      "offset": 0,
      "returnedCount": 1
   }
}
```

Get the Status of a Task

Method and URI:

```
GET https://api.ultradns.com/tasks/{taskId}
```

Parameters: Must include the specific Task ID.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Task DTO* in the body content.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If you do not have permission to read {taskId}.

Get the List of Tasks

Method and URI:

```
GET https://api.ultradns.com/tasks
```

Parameters: Can include the following:

Table 81 taskList Parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are *code* and *hasData*. Valid values for *code* are: ▪ **PENDING** ▪ **IN_PROCESS** ▪ **COMPLETE** ▪ **ERROR** Valid values for *hasData* are **TRUE** or **FALSE**. Default value of *hasData* is FALSE. Query operators need be followed by a **:** (colon). Example: • q="code:COMPLETE" • q="hasData:TRUE" • q="code:COMPLETE hasData:TRUE"	String.
offset	The position in the list for the first returned element (0 based). The default value is 0.	Integer.
limit	The maximum number of rows requested. The default value is 100.	Integer.
sort	The sort column used to order the list. Valid sort fields are CODE, CONTENT_TYPE, EXTENSIONS, HAS_DATA, and DATE. The default value is CODE.	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). The default value is false.	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *TaskList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to read tasks.

JSON Example: Response to Get List of Tasks

```
{
    "queryInfo": {
        "q": "\"code:COMPLETE hasData:FALSE\"",
        "sort": "DATE",
        "reverse": false,
        "limit": 5
    },
    "resultInfo": {
        "totalCount": 16,
        "offset": 0,
```

```
        "returnedCount": 5
    },
    "tasks": [
        {
            "taskId": "6b82d4ea-b983-4d1a-bdbb-9d6d40619fee",
            "code": "COMPLETE",
            "message": "Zone test-zone1.com. deleted."
        },
        {
            "taskId": "e440428a-b917-4373-9a5d-9ae9c9836027",
            "code": "COMPLETE",
            "message": "Zone test-zone2.com. deleted."
        },
        {
            "taskId": "d8bdb457-8aff-44b0-bf7a-159017b485cc",
            "code": "COMPLETE",
            "message": "Zone test-zone.com. deleted."
        },
        {
            "taskId": "15b360ba-534b-4ce4-b4bb-810bf6d9e8ef",
            "code": "COMPLETE",
            "message": "Zone test1-zone.com. deleted."
        },
        {
            "taskId": "fec42802-4512-49b3-bec9-6bf573a5f4cc",
            "code": "COMPLETE",
            "message": "Zone test2-zone.com. deleted."
        }
    ]
}
```


Get the Results of a Task

Method and URI

```
GET https://api.ultradns.com/tasks/{taskId}/result
```

Parameters: Must include a Task ID.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Task DTO* in the body content.

The content will be returned as a downloadable file. The name of the file will be the {taskId} that was submitted with the request. The file extension and content type are set by the background task and will be appropriate to the data returned.

Errors: An error is returned under the following conditions:

- If {taskId} does not exist.
- If you do not have permission for the task associated with the supplied {taskId}.
- If task is not yet completed.

Reporter Task DTOs

Reporter Task DTO

This describes the current state of a task.

Table 82 Task DTO

Field	Description	Type
taskId	Id for the task.	UUID.
code	Current state of the task.	Use one from: PENDING, IN_PROCESS, COMPLETE, ERROR.
message	Current message for the task.	String.
lastModifiedDateTime	Last Modified date time of the task.	Date-Time
resultUri	If task is COMPLETE, the URI from where the resulting data can be downloaded.	URI.

JSON Example: Task Status

```
{
    "taskId": "ZQV-013d3c5c-7b14-4ff0-b4af-453b76a827b6",
    "Code": "COMPLETE",
    "message": "Completed ZQV Report Successfully.",
    "lastModifiedDateTime": "2016-08-26T12:33:22.000Z",
    "resultUri": "https://api.ultradns.com/reports/tasks/ZQV-013d3c5c-7b14-
     4ff0-b4af-453b76a827b6"
}
```

Reporter TaskList DTO

The TaskList is returned when requesting the state of all tasks for a user.

Table 83 TaskList DTO

Field	Description	Type
tasks	The list of returned tasks. Each entry in the list matches the task DTO described above.	*Reporter Task DTO*.

JSON Example: Tasks

```
{
   "tasks": [
      {
         "taskId": "ZQV-013d3c5c-7b14-4ff0-b4af-453b76a827b6",
         "Code": "COMPLETE",
         "message": "Completed ZQV Report Successfully.",
         "lastModifiedDateTime": "2016-08-26T12:33:22.000Z",
         "resultUri": "https://api.ultradns.com/reports/tasks/ZQV-013d3c5c-
          7b14-4ff0-b4af-453b76a827b6"
      },
      {
```

```
        "taskId": "PQV-003f098b-a2df-437e-8482-65e8d93b4858",
        "Code": "COMPLETE",
        "message": "Completed PQVReport Successfully.",
        "lastModifiedDateTime": "2016-08-26T12:33:22.000Z",
        "resultUri": "https://api.ultradns.com/reports/tasks/PQV-003f098b-
        a2df-437e-8482-65e8d93b4858"
      }
   ]
}
```


Get the list of Tasks for Reporting

```
GET https://api.ultradns.com/tasks?taskType=reporting
```

Parameters:

Table 84 taskList Parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are *code* and *hasData*. Valid values for code are: ▪ PENDING ▪ IN_PROCESS ▪ COMPLETE ▪ ERROR Valid values for *hasData* are TRUE or FALSE. Query operators need be followed by a **:** (colon). Example: • q="code:COMPLETE” • q="hasData:TRUE" • q="code:COMPLETE hasData:TRUE"	String.
offset	The position in the list for the first returned element (0 based). The default value is 0.	Integer.
limit	The maximum number of rows requested. The default value is 100.	Integer.
sort	The sort column used to order the list. Valid sort fields are CODE, CONTENT_TYPE, EXTENSIONS, HAS_DATA, and DATE. The default value is CODE.	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). The default value is false.	Boolean.
taskType	For viewing reporting tasks user needs to set its value to "reporting". By default only Rest API configurations tasks will be returned as per current functionality.	String

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Reporter TaskList DTO* in the body content along with a *Link and Link Header*.

Errors: An error is returned under the following conditions:

- Invalid taskType
- No data is found.

Delete a Task

This call will remove information from the system for a task that has completed or is in an error-state.

Method and URI:

```
DELETE https://api.ultradns.com/tasks/{taskId}
```

Parameters: Must include a **Task ID**.

Body: None

Response: If delete completes, Status Code 204 is returned with no content in the body.

Errors: An error is returned under the following conditions:

- If you do not have permission to delete {taskId}.
- If {taskId} does not exist.
- If the task is not in a state that can be deleted.

Link and Link Header

This information can be seen in response header.

Table 85 Link and Link Header Response Information

Name	Type	Example
next	Link	</tasks?q="code:COMPLETE"&sort=DATE&taskType=reporting&offset=4&limit=2>; rel="next">
previous	Link	</tasks?q="code:COMPLETE"&sort=DATE&taskType=reporting&offset=0&limit=2>; rel="previous">
limit	Link Header	Initial request limit specified
results	Link Header	Total results in the current response

Web Forwards

UltraDNS Web Forwarding services allow you to redirect HTTP traffic from one target to another. Forwarding services are invaluable tools for companies that own or acquire multiple brands or have purchased many variations of their domain names to protect their brand identity.

Web Forwards DTOs

WebForward DTO

The WebForward DTO creates, modifies, or retrieves a Web Forward.

Table 86 Web Forward DTO

Attribute	Description	Type/ Restrictions
guid	System-generated unique identifier for this object.	Returned for *Get Web Forwards* call. Required for Update and Partial Update of Web Forwards.
requestTo	Specifies the URL to be redirected. The anchor character (#) is supported when creating a unique record. For example, *sub.abc.com/index.html* and *sub.abc.com/index.html#anchor* will be recognized and allowed.	Required for creation. Must be a valid URL. Only http:// is acceptable. The page portion of the URL is optional.
defaultRedirectTo	URL destination of the redirect.	Required for creation. Either http:// or https:// are acceptable.
defaultForwardType	Type of forward. Valid values include: ▪ Framed ▪ HTTP_301_REDIRECT ▪ HTTP_302_REDIRECT ▪ HTTP_303_REDIRECT ▪ HTTP_307_REDIRECT	Required for creation.
records	Present if you are using advanced web forward to specify where to forward, based on custom headers.	Array.

Attribute	Description	Type/ Restrictions
relativeForwardType	The Type of relative forward. Valid values include: ▪ **PARAMETER** – Parameter is appended to the target path. ▪ **PATH** – Path is appended to the target path. ▪ **PARAMETER_AND_PATH** – Both the Path and Parameter are appended to the target path.	String.
records/redirectTo	URL destination of the redirect.	Required on create (if records are present). Must be a valid URL. Can include a port number. Either http:// or https:// are acceptable.
records/forwardType	Type of forward. Valid values include: ▪ Framed ▪ HTTP_301_REDIRECT ▪ HTTP_302_REDIRECT ▪ HTTP_303_REDIRECT ▪ HTTP_307_REDIRECT	Required on create (if records are present).
records/priority	Order for a record to match. Lower numbers have higher priority.	Positive integer. Required on create (if records are present).
records/rules	Array of one or more rules.	Array Required on create (if records are present).
records/rules/header	Name of the header to match.	String Required on create (if records are present). Must be a header returned by the *Get Custom HTTP Headers of Account* call.
records/rules/matchCriteria	Type of match to perform. Valid values include: ▪ BEGINS_WITH ▪ CONTAINS ▪ ENDS_WITH ▪ MATCHES	Required on create (if records are present).

Attribute	Description	Type/ Restrictions
records/rules/value	Expected header value.	String Required on create (if records are present).
records/rules/caseInsensitive	Flag to indicate if the match takes case into account (**true**) or not (**false**).	Boolean If not present, defaults to false.

WebForwardList DTO

The webForwardList is returned for the list of web forwards.

Table 87 WebForwardList DTO

Attribute	Description	Type
webForwards/webForward	One of the returned webForwards. Structure matches the webForward DTO described above.	*Web Forward DTO*
queryInfo/q	The query used to construct the list.	String.
queryInfo/sort	The sort column used to order the list.	String.
queryInfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

Custom Header List DTO

Custom Header List returns the list of custom headers defined for an account.

Table 88 Custom header List DTO

Attribute	Description	Type
names	An array of the custom header names.	Array.

JSON Example: Custom Header List

```
{
    "names":
      [
        " ",...
      ]
}
```

Web Forward Create DTO

The REST API returns the guid when you create a web forward.

Table 89 Web Forward Create DTO

Attribute	Description	Type
guid	The unique identifier for the web forward.	String.

JSON Example: Web Forward Create

```
{
    "guid": "0909433CB37A13A8"
}
```

Add Custom HTTP Header to Account

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/customheaders
```

Parameters: None

Body: Must include a *Custom Header List DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you don't have permission to create custom headers.
- If you don't have permission to access the specified account or the account doesn't exist.
- If the custom header already exists.

Get Custom HTTP Headers of Account

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/customheaders
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Custom Header List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you don't have permission to access the specified account.
- If the account doesn't exist.

Delete Custom HTTP Header of Account

Method and URI:

```
DELETE
https://api.ultradns.com/accounts/{accountName}/customheaders/{headerName}
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no content in the body.

Errors: An error is returned under the following conditions:

- If you don't have permission to delete custom headers.
- If you don't have permission to access the specified account, or if the account doesn't exist.

Create Web Forwards

Method and URI:

```
POST https://api.ultradns.com/zones/{zoneName}/webforwards
```

Parameters: None

Body: Must include a *WebForward DTO*.

Response: If task completes, Status Code 201 is returned with a *Web Forward Create DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you don't have permission to create web forwards.
- If a web forwards already exsists for the <zone name>.

JSON Example: Create Web Forwards

```
{
    "requestTo": "a.demo-kb-1.com",
    "defaultRedirectTo": "https://b.demo-kb-1.co.us",
    "defaultForwardType": "HTTP_301_REDIRECT",
    "records": [
        {
            "redirectTo": "c.demo-kb-1.com.in",
            "forwardType": "HTTP_301_REDIRECT",
            "priority": 1,
            "rules": [
                {
                    "header":"Accept",
                    "matchCriteria": "CONTAINS",
                    "value": "kb",
                    "caseInsensitive": false
```

```
                }
            ]
        }
    ]
}
```

JSON Example: Create Web Forwards with anchor character (#)

In the following example, duplicate Web Forward records are created, however, one record is created using the anchor character (#). In this scenario, an error would not occur as the records are not duplicates due to the handling of the anchor character (#) making the record unique.

```
{
   "requestTo":"a.demo-kb-1.com/index.html",
   "defaultRedirectTo":"https://b.demo-kb-1.co.us",
   "defaultForwardType":"HTTP_301_REDIRECT"
}
{
   "requestTo":"a.demo-kb-1.com/index.html#anchor",
   "defaultRedirectTo":"https://b.demo-kb-1.co.us",
   "defaultForwardType":"HTTP_301_REDIRECT"
}
```

Get Web Forwards

Method and URI:

```
GET https://api.ultradns.com/zones/{zoneName}/webforwards
```

Parameters: You can include the following:

Table 90 Web Forwards Parameters

Parameter	Description	Type
q	The query used to construct the list. Query operators are: ▪ **type** - Valid values include: ○ Framed ○ HTTP_301_REDIRECT ○ HTTP_302_REDIRECT ○ HTTP_303_REDIRECT ○ HTTP_307_REDIRECT ○ Advanced ▪ **advanced** - Valid values include **true** and **false.** ▪ **name** – Valid values include any string, and will map to anything in either the host or the target.	String.
offset	The position in the list for the first returned element (0 based) Default value is 0.	Integer.
Limit	The maximum number of rows requested. Default value is 100.	Integer.

Parameter	Description	Type
sort	The sort column used to order the list. Valid sort values are: ▪ REQUEST_TO (this is the default) ▪ REDIRECT_TO ▪ TYPE ▪ DOMAIN ▪ ADVANCED	String.
reverse	Whether the list is ascending (**false**) or descending (**true**). Default value is false.	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *WebForwardList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you don't have permission to list web forwards.

JSON Example: Get Web Forwards

```
{
  "queryInfo": {
    "sort": "REQUEST_TO",
    "reverse": false,
    "limit": 100
  },
  "resultInfo": {
    "totalCount": 3,
    "offset": 0,
    "returnedCount": 3
  },
  "webForwards": [
    {
      "guid": "0909433CB37A13A8",
      "requestTo": "a.demo-kb-1.com",
      "defaultRedirectTo": "http://b.demo-kb-1.co.us",
      "defaultForwardType": "HTTP_301_REDIRECT",
      "records": [
        {
          "redirectTo": "http://c.demo-kb-1.com.in/",
          "forwardType": "HTTP_302_REDIRECT",
          "priority": 1,
          "rules": [
            {
              "header": "Accept",
              "matchCriteria": "CONTAINS",
              "value": "kb",
              "caseInsensitive": false
            }
          ]
        },
        {
          "redirectTo": "https://c.demo-kb-1.com.in/",
```

```
          "forwardType": "HTTP_302_REDIRECT",
          "priority": 2,
          "rules": [
            {
              "header": "Accept",
              "matchCriteria": "CONTAINS",
              "value": "kb",
              "caseInsensitive": false
            }
          ]
        }
      ]
    },
    {
      "guid": "0908433BB29E91EB",
      "requestTo": "www.demo-kb-1.com/community1/investors-foundation",
      "defaultRedirectTo": "http://b1.demo-kb-1.co.us",
      "defaultForwardType": "HTTP_301_REDIRECT"
    },
    {
      "guid": "0908433BB29D85C2",
      "requestTo": "www.demo-kb-1.com/community/investors-foundation",
      "defaultRedirectTo": "http://b1.demo-kb-1.co.us",
      "defaultForwardType": "HTTP_301_REDIRECT"
    }
  ]
}
```


Update Web Forward

Method and URI:

```
PUT https://api.ultradns.com/zones/{zoneName}/webforwards/{guid}
```

Parameters: None

Body: Must include a *WebForward DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you don't have permission to update web forwards.

Partially Update Web Forwards

The Partial Update Web Forward is a PATCH or JSON PATCH call and is generated as follows:

Method and URI:

```
PATCH https://api.ultradns.com/zones/{zoneName}/webforwards/{guid}
```

Parameters: None

Body: For standard XML or JSON calls, you must include a *WebForward DTO.*

For JSON PATCH formatted updates, the body must include a *JSON PATCH DTO*.

Patchable Objects for Web Forward:

- biz.neustar.ultra.rest.dto.WebForward

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If the {zoneName} is not valid.
- If you don't have permission to update web forwards.

Delete Web Forwards

Method and URI:

```
DELETE https://api.ultradns.com/zones/{zoneName}/webforwards/{guid}
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no content in the body.

Errors: An error is returned under the following conditions:

- If you don't have permission to delete web forwards.
- If the {guid} is not valid.

Extended Accounts API

The Extended Accounts API calls allow you to obtain additional information beyond the initial Accounts API created for the REST API. This new section includes information on various account-level objects such as: User Creation, Account Management, Security Group Management (Security Preferences, Security Questions, System Preferences), and User MetaData (for a current user).

This chapter provides details on the Accounts API calls available for use, as well as detailed Account DTO (Data Transfer Object) information. Where DTOs are required in the body of the call, or are returned as a response, cross reference links are provided to the specific table containing the details of DTO contents.

Create Zone Transfer Settings – Account Level

With this method, the zone transfer settings can be configured at the account level. Zone transfer settings include: Restrict IPs, TSIGs (transaction signature keys), and Notify Addresses.

When configured at the Account level, zone transfer settings, also referred to as Transfer ACL (Access Control List), are automatically inherited by every Primary zone belonging to the account that do not already have these items configured. They are also automatically applied to any new Primary zones created for the account.

Zone transfer settings can be changed at the zone level where appropriate, thereby overriding the account level settings. See *Create a Zone*, *Update a Zone*, and *Partially Update a Zone* sections of this guide for setting zone transfer restrictions at the zone level.

The Account-Level Zone Transfer settings calls have replaced the Add/Remove Restrict IPs for All Zones of Account API calls. If you have previously used those calls, please update your processes to use the new Transfer ACL calls described below.

Create Zone Transfer Settings is a POST call and is generated as follows:

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/transfer-acl HTTP/1.1
```

Parameters: None

Body: Must include a *Transfer ACL DTO.*

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body. Additionally, the timestamp of all active primary zones in the account will be updated.

Errors: An error is returned under the following conditions:

- You do not have permission to configure transfer settings for the specified {accountName}.
- If the {accountName} does not exist.
- If the provided Transfer ACL data is incorrect.

Update Zone Transfer Settings – Account Level

Updates to account level zone transfer settings (or Transfer ACL) are automatically inherited by every Primary zone owned by the account. Zone transfer settings include Restricted IPs, TSIGs (transaction signature keys), and Notify Addresses.

IMPORTANT: Because this is a full update, any Restrict IPs, TSIG, or Notify Addresses not included *will be deleted* from the Account level settings, and the deletions will be subsequently reflected in the zone transfer settings for all primary zones that inherit the account-level settings.

Update Zone Transfer Settings is a PUT call and is generated as follows:

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/transfer-acl HTTP/1.1
```

Parameters: None

Body: Must include a *Transfer ACL DTO*

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body. Additionally, the timestamp of all primary zones in the account that inherits account-level settings (that are subsequently updated with these changes) will be updated.

Errors: An error is returned under the following conditions:

- You do not have permission to configure transfer settings for the specified {accountName}.
- If the {accountName} does not exist.
- If the provided Transfer ACL data is incorrect. This includes restrictIP information where the range included for the update overlaps an existing IP address/range.

Partially Update Zone Transfer Settings – Account Level

Updates to account level zone transfer settings (or Transfer ACL) are automatically inherited by every Primary zone owned by the account. Zone transfer settings include Restricted IPs, TSIGs (transaction signature keys), and Notify Addresses.

Because this is a partial update, any Restrict IPs or Notify Addresses included will be appended to any existing settings, and TSIG will be updated if provided. These changes will subsequently be reflected to the Zone Transfer settings for all Primary zones that inherit account-level settings.

The Partial Update Zone Transfer Settings is a PATCH call and is generated as follows:

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/transfer-acl HTTP/1.1
```

Parameters: None

Body: Must include a *Transfer ACL DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body. Additionally, the timestamp of all primary zones in the account that inherit account-level settings (that are subsequently updated with these changes) will be updated.

Errors: An error is returned under the following conditions:

- You do not have permission to configure transfer settings for the specified {accountName}.
- If the {accountName} does not exist.
- If the provided Transfer ACL data is incorrect. This includes restrictIP information where the range included for the update overlaps an existing IP address/range.

Remove Zone Transfer Settings – Account Level

This call removes zone transfer settings from the account level. The removal of account-level zone transfer information is automatically passed to all active primary zones of the account that are configured to inherit account level zone transfer settings.

The Account-Level Zone Transfer settings calls have replaced the Add/Remove Restrict IPs for All Zones of Account API calls. If you have previously used those calls, please update your processes to use the new Transfer ACL calls described here.

The removal Zone Transfer Settings is a DELETE call and is generated as follows:

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/transfer-acl HTTP/1.1
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no content in the body. Additionally, the timestamp of all active primary zones in the account will be updated.

Errors: An error is returned under the following conditions:

- You do not have permission to configure transfer settings for the specified {accountName}.
- If the {accountName} does not exist.

Allowed IP Ranges – Account level

This provides the ability to limit access to the UltraDNS Portal or to the REST API, and to one or more enumerated ranges of IP addresses. If this is not set, then all IP addresses are valid. If it is set, then only clients within the specified IP ranges will be allowed to access the UltraDNS Portal or to REST API. The IP restrictions can be limited to just the UltraDNS Portal, just the API, or can apply to both.

If you accidentally block yourself from accessing both the UltraDNS Portal and the REST API, you will need to contact customer support and verify your identity before you are able to access any UltraDNS services.

Get All Allowed Account-Level IP Range

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/allowedips
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *AccountIPRangeList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to specify account-level IP restrictions.

JSON Example: GET All Allowed Account-Level IP Range

```
{
   "resultInfo": {
      "totalCount": 1,
      "offset": 0,
      "returnedCount": 1
      }

   "accountIPRanges": [
   {
      "guid": 0F0840B06DE2D354",
      "startIP": "1.1.1.1",
      "endIP": "2.2.2.2",
      "applications": [
         "UI"
         ]
      }
   ]
}
```

Add an Allowed Account-Level IP Range

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/allowedips
```

Parameters: None

Body: Must include an *AccountIPRange DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you do have permission to specify account-level IP restrictions.

JSON Example: Add an Allowed Account-Level IP Range

```
{
   "startIP": "10.10.10.10",
   "endIP": "20.20.20.20",
   "comments": "This is a comment",
   "applications": [
         "UI"
   ]
}
```

Update Allowed Account-Level IP Range

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/allowedips/{guid}
```

Parameters: None

Body: Must include an *AccountIPRange DTO*.

Response: If task completes, Status Code 200 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you do have permission to specify account-level IP restrictions.

JSON Example: Update an Allowed Account-Level IP Range

```
{
   "startIP": "11.11.11.11",
   "endIP": "18.18.18.18",
   "comments": "This is updated comment",
   "applications": [
         "UI"
   ]
}
```

Delete an Allowed Account-Level IP Range

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/allowedips/{guid}
```

Parameters: None

Body: None

Response: If delete completes completes, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- If it is not a valid {guid}.
- If you do not have permission to specify account-level IP restrictions.

Accounts API

The Accounts API calls allow you to obtain information on various account-level objects as well as manage the Zone Transfer information for the account.

This chapter provides details on the Accounts API calls available for use, as well as detailed Account DTO (Data Transfer Object) information. Where DTOs are required in the body of the call, or are returned as a response, cross reference links are provided to the specific table containing the details of DTO contents.

GET Accounts of a User

Provides a list of all accounts to which the user is assigned. The “user” is assumed to be the user whose credentials are currently being used for API call authentication.

Method and URI:

```
GET https://api.ultradns.com/accounts
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Account List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to view accounts.

JSON Example: Account List DTO

```
{
      resultInfo: {
         totalCount: 1
         offset: 0
         returnedCount: 1
   }
      accounts: [
        {
         "accountName": "WidgetEng",
         "accountHolderUserName": "Widget Engineering",
         "ownerUserName": "AlphaEngineer",
         "numberOfUsers": 15,
         "numberOfGroups": 3,
         "accountType": "ORGANIZATION"
         "features": [
               "ADVDIRECTIONAL",
               "DNSSEC",
```

```
                "MAILBACKER",
                "RECURSIVE",
                "REPORTING",
                "RNAME",
                "SITEBACKER",
                "TRAFFICCNTRL",
                "WEBFORWARD"
                ]
            }
        ]
}
```

GET a Single Account

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Account List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to view the account.

GET Zones of an Account

This call provides a summarized list of Zones owned by an accountName. You can return zones either by **type** or by **name** (including partial name matches).

The Get Zones of Account call is a GET call and is generated as follows:

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/zones
```

Parameters: You can include the following:

Table 91 Parameters for Get Zones of an Account

Parameter	Description	Type
q	The query used to construct the list. Query operators are: ▪ **name** – Name of the zone (allowing for partial string matches). ▪ **zone_type** – Type of zone you want to be listed. If not specified, all zone types are returned. Use one of the following values: ○ ALIAS ○ PRIMARY ○ SECONDARY	String.
offset	The position in the list for the first returned element (0 based). Default is 0.	Integer.

Parameter	Description	Type
limit	The maximum number of rows requested. Default is 100.	Integer.
sort	The sort column used to order the list. The valid values are: ▪ NAME (default) ▪ ZONE_TYPE	String.
reverse	List is sorted in Ascending order by default, with parameter value being **false**. Enter **true** to sort the list in Descending order by the sort column specified (or by Name if no sort value is entered).	Boolean.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Zone List DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to access the specified {accountName}.
- If you do not have permission to view zones in the {accountName}.

JSON Example: Get Zones of an Account Response


```
{
    "queryInfo": {
        "sort": "NAME",
        "reverse": false,
        "limit": 100
    },
    "resultInfo": {
        "totalCount": 5530,
        "offset": 0,
        "returnedCount": 100
    },
    "zones": [
        {
            "properties": {
                "name": "teamtest1.com.",
                "accountName": "team",
                "type": "PRIMARY",
                "dnssecStatus": "UNSIGNED",
                "status": "ACTIVE",
                "owner": "team",
                "resourceRecordCount": 33,
                "lastModifiedDateTime": "2018-06-22T13:59Z"
            },
            "registrarInfo": {
                "nameServers": {
                    "missing": [
                        "test.ourtest1.net",
                        "test.ourtest2.net."
                    ]
                }
```

```
            },
            "inherit": "ALL"
        },
        {
            "properties": {
                "name": "teamtest2.com.",
                "accountName": "team",
                "type": "PRIMARY",
                "dnssecStatus": "UNSIGNED",
                "status": "ACTIVE",
                "owner": "team",
                "resourceRecordCount": 13,
                "lastModifiedDateTime": "2018-06-25T09:22Z"
            },
            "registrarInfo": {
                "nameServers": {
                    "missing": [
                        "test.ourtest3.net.",
                        "test.ourtest4.net"
                    ]
                }
}
```

GET Users of an Account

This call returns a list of all of the Users assigned to a specified accountName, which are sorted by user name in ascending order.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/users
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *User List DTO* in body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to access the specified {accountName}.
- If you do not have permission to view users in the {accountName}.

JSON Example: User List DTO

```
{
  "resultInfo": {
    "totalCount": 10,
    "offset": 0,
    "returnedCount": 10
  },
  "users": [
    {
      "userName": "JTDoe",
```

```
      "firstName": "John",
      "lastName": "Doe",
      "primaryEmail": "jtdoe@neustar.biz",
      "secondaryEmail": "jtdoe@team.neustar",
      "phone": "13333333333",
      "fax": "14444444444",
      "mobile": "12222222222",
      "companyName": "Neustar",
      "apiOnlyUser": false,
      "authType": "SHA1"
    },
    {
      "userName": "JPRoe",
      "firstName": "Jane",
      "lastName": "Roe",
      "primaryEmail": "jproe@neustar.biz",
      "secondaryEmail": "jproe@team.neustar",
      "phone": "15555555555",
      "fax": "16666666666",
      "mobile": "17777777777",
      "companyName": "Neustar",
      "apiOnlyUser": true,
      "authType": "SHA1"
    }
  ]
}
```


Response of this call can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

CSV Example: List of users response in .CSV format


```
Status 200 OK
[
userName,firstName, lastName, primaryEmail, secondaryEmail, phone, fax,
mobile, companyName, apiOnlyUser, authType
JTDoe,John,Doe,jtdoe@neustar.biz,jtdoe@team.neustar,13333333333,14444444444,1
2222222222,Neustar,false, SHA1
JPRoe,Jane,Roe,jproe@neustar.biz,jproe@team.neustar,15555555555,16666666666,1
7777777777,Neustar,true, SHA1
]
```


Delete Access of a User from an Account

This call removes the access of a User that is assigned to a specified Account. **Note: If the designated user is only assigned to one account, this call will delete the user.**

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/users/{userName}
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- Invalid User {username}.
- Invalid Account {accountName}.
- If the target account is not currently in an Active state.
- If you do not have permission to perform the delete for the account.
- User is no longer affiliated to the Account.

Account DTOs

The sections and tables below, provide detailed information about the contents of the DTOs used for Account API calls. Where a DTO field consists of the contents of another DTO, a cross reference link to the associated DTO is provided. When possible, return links to the "parent" DTO are also provided, along with links to the API calls that use the DTO.

Account DTO

This call provides summary information about the Account. This is integrated into the *Account List DTO*, which is returned by the *Get Accounts of User* call.

Table 92 Account DTO

Field	Description	Type	Editable in PUT/PATCH
accountName	Name of the account.	Name.	Yes
accountHolderUserName	User name for the account holder.	Name.	No
ownerUserName	User name of the owner (primary user).	Name.	Yes
ownerAddress	Address of the owner (primary user).	*Address DTO*	Yes
accountHolderAddress	Address of the account holder. (This field will be returned and can be updated only for the ORGANIZATION account type)	*Address DTO*	Yes (only for accounts of type ORGANIZATION) . If a user tries to update the accountHolderAddress for non-ORGANIZATION account types, a validation error will be returned.
numberOfUsers	User(s) count/quantity for account.	Integer.	No

Field	Description	Type	Editable in PUT/PATCH
numberOfGroups	Group count/quantity for account.	Integer.	No
accountType	Type of the account.	One from: ▪ INDIVIDUAL ▪ ORGANIZATION	No
features	List of returned features per account, that can include: ▪ ADVDIRECTIONAL ▪ DNSSEC ▪ LAB_ENABLED ▪ MAILBACKER ▪ MDDI ▪ RECURSIVE ▪ REPORTING ▪ RNAME ▪ SITEBACKER ▪ TRAFFICCNTRL ▪ WEBFORWARD	String.	No
accountId	The short name of the account. (This is found from the ACCT_NBR column from the BILLING_ACCOUNT table).	Name.	No
status	The status of the account.	One from: ▪ Active ▪ Suspended	No
created	Date the account was created in the ISO 8601 format.	Date.	No
defaultSOAEmail	The default value used for the SOA email address in newly created zones. If null, not present.	Email.	Yes
restrictAccessIPs	List of IP restrictions that apply for all users in the account.	List of *RestrictAccessIP DTO.*List of *Table 115 RestrictAccesIP DTO.*	Yes
accountNameServers	The list of the Account Name Servers, along with the state of the Name Server.	List of AccountNameServer DTO.	No

Field	Description	Type	Editable in PUT/PATCH
usageLimit	The maxiumum number of various record types, queries, and/or pools that are configured for your account.	List of *UsageLimit DTO*.	No

JSON Example: Account DTO Body

```
{
   "accountName": "sample",
   "accountHolderUserName": "sampleUser",
   "ownerUserName": "sampleUser",
   "numberOfUsers": 1,
   "numberOfGroups": 3,
   "accountType": "ORGANIZATION",
   "accountID": "AG02-200",
   "status": "ACTIVE",
   "created": "20150101T12:00Z",
   "defaultSOAEmail": "test@ultradns.com",
   "accountHolderAddress": {
        "address1": "address1",
        "address2": "address2",
        "country": "USA",
        "state": "Texas",
        "city": "Indore",
        "zip": "452010"
   },
   "accountNameServers": [
       {
            "nameServer": "ns1.pdns.com.",
            "state": "Active"
       },
       {
            "nameServer": "ns2.pdns.com.",
            "state": "Active"
       }
   ],
   "usageLimit": {
   "sitebackerRecordsLimit": 3000,
   "domainsLimit": 900,
   "recordsLimit": 8900,
   "queriesLimit": 808090,
   "webForwardsLimit": 24354,
   "trafficControllerRecordsLimit": 7868,
   "directionalPoolsLimit": 7979
  }
}
```

Account List DTO

Table 93 Account List DTO

Field	Description	Type
accounts/account	One of the returned accounts. The structure should match the *Account DTO*.	*Account DTO*.
features	List of returned features per account, that can include: ▪ ADVDIRECTIONAL ▪ DNSSEC ▪ LAB_ENABLED ▪ MAILBACKER ▪ MDDI ▪ RECURSIVE ▪ REPORTING ▪ RNAME ▪ SITEBACKER ▪ TRAFFICCNTRL ▪ WEBFORWARD	String.
queryinfo/q	The query used to construct the list.	String.
queryinfo/sort	The sort column used to order the list.	String.
queryinfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryinfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: Account List DTO Body

```
{
   "accounts": [
      {
         "accountName": "sample",
         "accountHolderUsername": "sampleUser",
         "ownerUserName": "sampleUser",
         "numberOfUsers": 1,
         "numberOfGroups": 1,
         "accountType": "ORGANIZATION",
      }
   ]
   "queryInfo": {
         "q": " ",
         "sort": " ",
         "reverse": "TRUE",
         "limit": 2
      }
```

```
   "resultInfo": {
         "totalCount": 2,
         "offset": 0,
         "returnedCount": 2
      }
}
```

AccountNameServer DTO

Table 94 AccountNameServer DTO

Field	Description	Type
nameServer	The Account Name Server.	String.
ipv4Address	An IPV4 address.	String.
ipv6Address	An IPV6 address.	String.
state	The Record state (i.e. Active or Pending).	String.

UsageLimit DTO

Table 95 Usage Limit DTO

Field	Description	Type
sitebackerRecords	Usage limit of querying sitebacker records.	Integer
zones	Usage limit of queryingZones.	Integer
records	Usage limit of querying resource records.	Integer
queries	Usage limit of querying any DNS-Queries.	Integer
webForwards	Usage limit of querying web forward records.	Integer
trafficControllerRecords	Usage limit of querying traffic controller records.	Integer

User DTO

Table 96 User DTO

Field	Description	Type
userName	User name in UltraDNS system.	Name
firstName	User's given name.	String
lastName	User's family name.	String
primaryEmail	The main email address for the user.	Email
secondaryEmail	The backup email address for the user. Optional.	Email
phone	Phone number.	String
fax	Fax number. Optional.	String
mobile	Cell phone number. Optional.	String

Field	Description	Type
companyName	Name of the company.	String
apiOnlyUser	Displays if the designated user ONLY has API access (true), or of the user has both UI AND API access (false).	Boolean
authType	Displays the User Authentication type. Optional.	String

JSON Example: User DTO Body

```
{
   "userName": "JTDoe",
   "firstName": "John",
   "lastName": "Doe"
}
```

User List DTO

Table 97 User List DTO

Field	Description	Type
users/user	One of the returned users. The structure will match the *User DTO*.	List of *User DTO.*
queryinfo/q	The query used to construct the list.	String.
queryinfo/sort	The sort column used to order the list.	String.
queryinfo/reverse	Whether the list is ascending (**false**) or descending (**true**).	Boolean.
queryInfo/limit	The maximum number of rows requested.	Integer.
resultInfo/totalCount	Count of all zones in the system for the specified query.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer.
resultInfo/returnedCount	The number of records returned.	Integer.

JSON Example: UserList DTO Body

```
{
        "users" : [
            {
                "userName": "JTDOE",
                "firstName": "John",
                "lastName": "Doe"
            }
        ],
        "queryInfo": {
            "q": " ",
            "sort": " ",
            "reverse": "TRUE",
            "limit": 2
```

```
        },
        "resultInfo": {
            "totalCount": 2,
            "offset": ,
            "returnedCount": 2
        }
}
```

Address DTO

Table 98 Address DTO

Field	Description	Type
address1	The first line of the address.	String.
address2	The second line of an address if necessary. Optional.	String.
country	Country the address resides in.	String. Validated using the ISO-3-661 two letter codes for countries.
state	The state or province the address resides in. Optional if outside of the United States or Canada.	String. Validated using the ISO-3166-2: US standard for US States and territories, and the ISO-3661-2: CA Standard for Canadian provinces and territories.
city	The city in which the address resides.	String.
zip	The zip code / postal code for the address. Optional	String.

User MetaData

Transfer ACL DTO

The Transfer ACL DTO contains zone transfer information to be set at the account level. This DTO is required in the body of the *Create Zone Transfer Settings – Account Level*, *Update Zone Transfer Settings – Account Level*, and *Partially Update Zone Transfer Settings – Account Level*.

Zone transfer settings can also be changed or removed at the zone level where appropriate. See *Create a* Zone, and *Partially Update a Zone* sections of this guide for information on configuring zone transfer settings at the zone level.

Table 99 Transfer ACL DTO

Field	Description	Type
restrictIPList	The list of IP ranges that are allowed to use AXFR to transfer primary zones out.	List of *Restrict IP DTOs.*

Field	Description	Type
tsig	The TSIG information for the primary zone.	*TSIG DTO.*
notifyAddresses	The addresses that are notified when updates are made to the primary zone.	List of *Notify Address DTOs.*

AccountIPRange DTO

Table 100 AccountIPRange DTO

Field	Description	Type
guid	The internal ID for range. If specified during creation, it is ignored. It is returned when getting the list of all allowed IP ranges.	String.
startIP	The first allowed IP address in the range, inclusive.	Valid ipv4 address.
endIP	The last allowed IP address in the range, inclusive.	Valid ipv4 address.
comments	Optional comments.	String.
applications	The list of applications that this range applies to. There must be at least one value specified. Valid values are: ▪ **UI** – When included in the list, the allowed IP range applies to the UltraDNS Portal. ▪ **API** – When included in the list, the allowed IP range applies to the REST API.	List of one or more valid values.

AccountIPRangeList DTO

Table 101 AccountIPRangeList DTO

Field	Description	Type
accountIPRanges	A list of *AccountIPRange DTO*.	List of *AccountIPRange DTO.*
resultInfo/totalCount	The number of accountIPRanges.	Integer.
resultInfo/offset	The starting point for the list. This is currently always 0 for AccountIPRangeList, as pagination is not currently supported.	Integer.
resultInfo/returnedCount	The number of accountIPRange DTOs returned. This is currently equal to the total count, as pagination is not supported.	Integer.

Get Address Info for a Current User

Method and URI:

```
GET https://api.ultradns.com/address
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Address DTO* in the body content.

Errors: None

JSON Example: Get Address Info of Current User Response

```
{
    "address1": "asis",
    "address2": "add222",
    "country": "UGA",
    "state": "Va",
    "city": "fsff",
    "zip": "20147"
}
```

Update Address Info for a Current User

Method and URI:

```
PUT https://api.ultradns.com/address
```

Parameters: None

Body: Must contain an *Address DTO*

Response: If task completes, Status Code 200 is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If invalid data was submitted in the body.

Get Details of Current User

Method and URI:

```
GET https://api.ultradns.com/user
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 is returned with User DTO in response body.

Errors: None

JSON Example: Get details of Current User Response

```
{
   "userName": "JTDoe",
   "firstName": "John",
   "lastName": "Doe",
   "primaryEmail": "test1@ultradns.com",
   "secondaryEmail": "test2@ultradns.com ",
   "phone": "123456789",
   "fax": "123456789",
   "mobile": "123456789",
   "companyName": "Neustar"

}
```

Update Details of Current User

Method and URI:

```
PUT https://api.ultradns.com/user
```

Parameters: None

Body: Must include the User DTO.

Response: If task completes, Status Code 200 is returned with User DTO in response body.

Errors: An error is returned under the following conditions:

- If there is an attempt to update the Username.

User Creation

This API call allows you to add a new user to the REST API, or to re-invite a user that did not receive the initial user creation invitation. You will need to know the security group name before you can create the invitation for the user.

Invite New User

Method and URI:
`POST https://api.ultradns.com/accounts/{accountName}/users`

Parameters: None

Body: Must contain the *UserInvite DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you have permission to list users in the account, but do not have permission to invite new users.
- If you do not have permission to access the account.
- If the account is not currently in an Active state.
- If invalid data was submitted in the UserInvite DTO (invalid/missing email address, invalid/missing security group).

JSON Example: Add Non-API User to Standalone Group

```
{
   "email": "email@email.com",
   "isApiOnlyUser": false,
   "isStandalone": true
}
```

JSON Example: Add Non-API User to Group

```
{
   "email": "email@email.com",
   "isApiOnlyUser": false,
   "isStandalone": false,
   "group": "TECHNICAL"
}
```

Re-Invite User

Method and URI:
`PUT https://api.ultradns.com/accounts/{accountName}/users`

Parameters: None

Body: Must contain the *UserInvite DTO*.

Response: If task completes, Status Code 200 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you have permission to list users in the account, but do not have permission to invite new users.
- If you do not have permission to access the account.
- If invalid data was submitted in the UserInvite DTO (invalid/missing email address, invalid/missing security group).

Get Pending Invitations

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/users/invitation
```

Parameters: None

Body: Must contain the *UserInviteList DTO*.

Response: If task completes, Status Code 200 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you do not have permission to access the account.

JSON Example: Get Pending User Invite Invitations

```
{
    "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
    },
    "invitations": [
        {
            "group": "Technical",
            "isStandalone": false,
            "isApiOnlyUser": false,
            "email": "email@yahoo.com"
        }
    ]
}
```

Delete Pending Invitations

This API is only accessible for the owner or administrative user of the account.

Method and URI:

```
DELETE
https://api.ultradns.com/accounts/{accountName}/users/invitation/{emailAddress}
```

Parameters: None

Body: None

Response: If task completes, Status Code 204 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- If you do not have permission to access the account.
- If emailAddress does not exist as a user invitation.

UserInvite DTO

Table 102 UserInvite DTO

Attribute	Description	Type
group	Name of the security group for the user.	String. Valid security group for account. Required for Invite User when isStandalone is “False.” Ignored for Re-Invite User and when isStandalone is “true.”
isStandalone	Whether or not the user is a standalone user. If set to **true**, user will not belong to any group. If set to **false**, user will be assigned to the group specified in the **group** attribute.	Boolean. Required. Defaults to false. Ignored for Re-Invite User.
isApiOnlyUser	Whether or not the user has api only access. If set to **true**, user will only be able to access the API services. If set to **false**, user will be able to access both the API services and UltraDNS Managed Services Portal.	Boolean. Defaults to false. Ignored for Re-Invite User.
email	Email address for the new user.	Email. Required.

UserInviteList DTO

Table 103 UserInviteList DTO

Attribute	Description	Type
resultInfo/totalCount	The total count of all pending user invitations.	Integer.
resultInfo/offset	The position in the list for the first returned element. (0 based)	Integer. Always 0.

Attribute	Description	Type
resultInfo/returnedCount	The number of records returned.	Integer.
invitations	List of *UserInvite DTO*.	*UserInvite DTO*

Account Management

Get Account Info

Method and URI:
`GET https://api.ultradns.com/accounts/(accountName)`

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *Account DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you do not have permission to access the account.

JSON Example: Get Account Info Response for INDIVIDUAL Account type

```
{
    "accountName": "team",
    "accountHolderUserName": "teamtest",
    "ownerUserName": "teamtest",
    "ownerAddress": {
        "address1": "asis",
        "address2": "add222",
        "country": "UGA",
        "state": "Va",
        "city": "fsff",
        "zip": "20147"
    },
    "numberOfUsers": 2,
    "numberOfGroups": 16,
    "accountType": "INDIVIDUAL",
    "accountId": "ABC-1234",
    "status": "ACTIVE",
    "created": "2017-06-09T13:40Z",
    "features": [
        "ADVDIRECTIONAL",
        "DNSSEC",
        "LAB_ENABLED",
        "MDDI",
        "RECURSIVE",
        "REPORTING",
        "SITEBACKER",
        "TRAFFICCNTRL",
        "WEBFORWARD",
        "ZBR"
    ]
}
```

JSON Example: Get Account Info Response for ORGANIZATION Account type

```
{
    "accountName": "team",
```

```
    "accountHolderUserName": "teamtest",
    "ownerUserName": "teamtest",
    "ownerAddress": {
        "address1": "asis",
        "address2": "add222",
        "country": "UGA",
        "state": "Va",
        "city": "fsff",
        "zip": "20147"
    },
    "accountHolderAddress": {
        "address1": "address1",
        "address2": "add222",
        "country": "UGA",
        "state": "Va",
        "city": "fsff",
        "zip": "20147"
    },
    "numberOfUsers": 2,
    "numberOfGroups": 16,
    "accountType": "ORGANIZATION",
    "accountId": "ABC-1234",
    "status": "ACTIVE",
    "created": "2017-06-09T13:40Z",
    "features": [
        "ADVDIRECTIONAL",
        "DNSSEC",
        "LAB_ENABLED",
        "MDDI",
        "RECURSIVE",
        "REPORTING",
        "SITEBACKER",
        "TRAFFICCNTRL",
        "WEBFORWARD",
        "ZBR"
    ]
}
```


Update Account Info

This method allows a user to change the primary user of an account, change the primary user's address information, change the account name, or change the SOA email address.

Method and URI:

```
PUT https://api.ultradns.com/accounts/(accountName)
```

Parameters: None

Body: Must incude an *Account DTO*.

Since this is a PUT method, all fields that can be edited by the user must be specified. The read-only fields in the Account DTO are optional and do not need to be specified. Any non-

specified optional fields will be assumed to have an empty value, which will replace any already assigned values.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.
- If you have permission to get the account info, but do not have permission to edit the account info.

  If you are not allowed to edit some of the fields that are specified in your input, a 403 error code will be returned, and none of the edits will take effect.
- If you do not have permission to access the account.

Partially Update Account Info

This method allows a user to change the primary user of an account, change the primary user's address information, change the account name, or change the SOA email address.

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}
```

Parameters: None

Body: Must include an *Account DTO*.

Since this is a PATCH method, only the fields being modified need to be specified. Any fields that are not specified will retain their already assigned value(s).

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.
- If you have permission to get the account info, but do not have permission to edit the account info.
- If you are not allowed to edit some of the fields that are specified in your input, a 403 error code will be returned, and none of the edits will take effect.
- If you do not have permission to access the account.

TTL DTO

Table 104 TTL DTO

Attribute	Description	Type
type	The type of record whose TTL is being set.	One of the following resource record types: ▪ A ▪ AAAA ▪ CNAME ▪ MX ▪ TXT ▪ SRV ▪ NS ▪ PTR ▪ RP ▪ HINFO ▪ NAPTR ▪ SOA ▪ SPF Also supported are these special cases: ▪ ANY (All resource records) ▪ SBTC (Sitebacker and Traffic Controller pools) ▪ SOA_REFRESH (Refresh field in SOA) ▪ SOA_RETRY (Retry field in SOA) ▪ SOA_EXPIRE (Expire field in SOA) ▪ SOA_MIN_CACHE (Min Cache field in SOA)
defaultValue	The default value for the TTL.	Integer. Between 0 and 2147483647
min	The minimum value for the TTL.	Integer. Between 0 and 2147483647
max	The maximum value for the TTL.	Integer. Between 0 and 2147483647

JSON Example: TTL DTO

```
{
   "type": "A",
   "defaultValue": 123,
   "min": 12,
   "max": 512
}
```

TTLList DTO

Table 105 TTLList DTO

Attribute	Description	Type
ttls	The specified TTLs for the account.	List of *TTL DTO*.

Get Account TTLs

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/ttls
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *TTLList DTO* in the body content.

Errors: An error is returned under the following conditions:

- If you have permission to get the account info, but do not have permission to edit the account info.
- If you do not have permission to access this account.

JSON Example: Get Account TTLs Response

```
{
    "ttls": [
        {
            "type": "MX",
            "max": 1111
        },
        {
            "type": "NAPTR",
            "defaultValue": 9010
        },
        {
            "type": "TXT",
            "defaultValue": 100,
            "min": 30,
            "max": 1000
        },
        {
            "type": "TLSA",
            "defaultValue": 222
        },
        {
            "type": "A",
            "defaultValue": 500
        }
    ]
}
```

Update Account TTLs

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/ttls
```

Parameters: None

Body: Must include a *TTL DTO*.

Since this is a PUT function, all TTLs and all the fields in the TTL DTOs must be specified, otherwise the values will be reset to the default (empty) value.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.
- If you have permission to get the account info, but do not have permission to edit the account info.
- If you do not have permission to access this account.

Partially Update Account TTLs

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/ttls
```

Parameters: None

Body: Must include a *TTL DTO*.

Since this is a PATCH function, only the TTLs being updated need to be included in the list. All TTLs and fields in the TTL DTOs not included in the list will retain their current state.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.
- If you have permission to get the account info, but do not have permission to edit the account info.
- If you do not have permission to access this account.

Security Group Management

Security Group DTO

Table 106 Security Group DTO

Attribute	Description	Type
name	The name of the security group. Returned on a GET call. Ignored on a POST/PUT/PATCH call.	String.
entries	The security group entries for this security group.	List of *Security Group Entry DTO.*
exceptions	A list of security exceptions.	*Security Exception List DTO*
usersCount	The total number of users in the group(s) being returned. Returned on a GET call. For Standalone groups, the usersCount will be reflected in the returned totalCount value, as only one user can be assigned to a Standalone group.	Integer.

JSON Example: Security Group DTO

```
{
   "name": "1441307959243group2",
   "entries":[
      {
         "type": "ACCOUNT",
         "permission": "READ"
      }
   ]
}
```

Security Group Entry DTO

Table 107 SecurityGroupEntry DTO

Attribute	Description	Type
type	The name of the Object or Page being configured. The specific **Permission** must to be set to a record type if the **Permission** is being set at the pool level. The Permission and Type attributes work as a Top-Down attribute.	String. Required to have one of the following values: ▪ ACCOUNT ▪ APEXALIAS ▪ DOMAIN_SERVICES ▪ ZONE ▪ RESOURCE_RECORDS

Attribute	Description	Type
	For example: If the DELETE permission is set for a SiteBacker Pool (that contains an A Record), then the DELETE permission must also be set for the A Record as well to allow changes to be made at the record level.	▪ A ▪ AAAA ▪ CAA ▪ CNAME ▪ DIR_POOL ▪ HINFO ▪ MX ▪ NAPTR ▪ NS ▪ PTR ▪ RP ▪ RD_POOL ▪ SB_POOL ▪ SRV ▪ SSHFP ▪ SPF ▪ TC_POOL ▪ TLSA ▪ TXT ▪ WEB_FORWARD ▪ REPORTS ▪ ACCOUNTS_PERMISSIONS ▪ ACCOUNT_PREFERENCES ▪ BILLING ▪ SERVICE_PACKAGE
permission	The permission being applied to the type. Permissions are cumulative: ▪ If you have Write, you also have Read, ▪ If you have Create, you also have Write and Read. ▪ etc.	String. Required to have one of the following values: ▪ NONE ▪ INHERIT ▪ READ ▪ WRITE ▪ CREATE ▪ DELETE ▪ GRANT GRANT is only legal when type is set to ACCOUNT. It is returned for the account owner, but cannot be given to another security group. SERVICE_PACKAGE is only allowed to have NONE, READ or INHERIT.

Attribute	Description	Type
		ACCOUNTS_PERMISSIONS, ACCOUNT_PREFERENCES, and BILLING are only allowed to have NONE, READ, WRITE, or INHERIT. ACCOUNT is not allowed to have NONE or INHERIT.
inheritedValue	Shows the calculated value for this security group entry if the permission is set to INHERIT. Returned on GET, ignored on POST/PUT/PATCH.	String. Only returned if permission is set to INHERIT. Must have one of the following values: ▪ NONE ▪ READ ▪ WRITE ▪ CREATE ▪ DELETE

Security Group List DTO

Table 108 SecurityGroupList DTO

Attribute	Description	Type
groups	A list of security groups.	List of SecurityGroup DTO objects.
resultInfo/totalCount	Count of all the exceptions returned.	Integer.
resultInfo/offset	The position in the list for the first returned element (0 based).	Integer. Always 0, since the pagination is not currently supported.
resultinfo/returnedCount	The number of records returned.	Integer.

Security Exception DTO

Table 109 SecurityException DTO

Attribute	Description	Type
groupName	The name of the group. This is only populated when getting the list of all exceptions for an object. For STANDLAONE groups, the groupName will be returned as the username, as opposed to a FirstName Lastname result.	String.
type	The type of the entity with the exception.	String. Required to have one of the following: ▪ ZONE

Attribute	Description	Type
		▪ A ▪ AAAA ▪ APEXALIAS ▪ CAA ▪ CNAME ▪ DIR_POOL ▪ HNFO ▪ MX ▪ NAPTR ▪ NS ▪ PTR ▪ RP ▪ RD_POOL ▪ SB_POOL ▪ SRV ▪ SSHFP ▪ SPF ▪ TC_POOL ▪ TLSA ▪ TXT ▪ WEB_FORWARD
name	The name of the entity.	String. Required: ▪ For a web forward, will be a **GUID.** ▪ For a mail forward, will be an **email address.** ▪ For a zone, it will be the **zone name** (with or without the trailing dot). ▪ For a resource record set or pool, it will be the **fully qualified domain name** for the rrset or pool (with or without the trailing dot).
permission	The permission being applied to the type. Permissions are cumulative. For Example: ▪ If you have Write, you also have Read. ▪ If you have Create, you also have Write and Read. ▪ Etc…	String. Required (for pools) to have one of the following: ▪ NONE ▪ READ ▪ WRITE ▪ CREATE ▪ DELETE
rdata	The Rdata of the Resource Record on which the exception should be set.	List <string>. Optional

Attribute	Description	Type
rrtype	The Resource Record type of the member record of the pools. It is mandatory for pools, but will be ignored for the following: Web Forward, Zone and Resource Records.	String. Required (for pools) to have one of the following: ▪ A ▪ AAAA ▪ CNAME ▪ HINFO ▪ MX ▪ NAPTR ▪ NS ▪ PTR ▪ RP ▪ SRV ▪ TXT

Security Exception List DTO

Table 110 SecurityExceptionList DTO

Attribute	Description	Type
exceptions	A list of security exceptions.	List of *Security Exception DTO* objects.
resultInfo/totalCount	The count of all exceptions returned.	Integer.
resultInfo/offset	The position in the list for the first returned element. (0 based)	Integer. Always 0.
resultInfo/returnedCount	The number of records returned.	Integer.

Create Security Group

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
```

Parameters: None

Body: Must include a *Security Group DTO*.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- Invalid data was submitted in the body, or the group name is already in use.

- If you have permission to get the account info, but do not have permission to create security groups.
- If you do not have permission to access this account.

JSON Example: Create Security Group without Exceptions

```
{
  "entries": [
                  {
                   "type": "ACCOUNT",
                   "permission": "CREATE"
                   },
                  {
                   "type": "RESOURCE_RECORDS",
                   "permission": "DELETE"
                   },
                  {
                   "type": "DOMAIN_SERVICES",
                   "permission": "READ"
                  },
                  {
                   "type": "ZONE",
                   "permission": "READ"
                  },
                  {
                   "type": "REPORTS",
                   "permission": "READ"
                  },
                  {
                   "type": "ACCOUNTS_PERMISSIONS",
                   "permission": "READ"
                  },
                  {
                   "type": "ACCOUNT_PREFERENCES",
                   "permission": "READ"
                  },
                  {
                   "type": "BILLING",
                   "permission": "READ"
                  },
                  {
                   "type": "SERVICE_PACKAGE",
                   "permission": "READ"
                  }
               ]
}
```

JSON Example: Create Security Group with Exceptions

```
{
  "entries": [
                  {
                   "type": "ACCOUNT",
                   "permission": "DELETE"
                   }
```

```
                ],
  "exceptions":
                {
                   "exceptions":
                           [
                              {
                              "type": "ZONE",
                              "permission": "CREATE",
                              "name": "zoneName"
                              },
                              {
                              "type": "DIR_POOL",
                              "permission": "WRITE",
                              "name": "DRR.AAAA-test.com",
                              "rrtype": "A",
                              "rdata": ["{1.1.1.1}"]
                              },
                               {
                               "type": "RD_POOL",
                               "permission": "CREATE",
                               "name": "rd.aaaa-test.com. ",
                               "rrtype": "A"
                               }
                           ]
                 }
}
```

Get a Security Group

```
GET https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Security Group DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to DNSSECview the security groups.
- If the Group does not exist, the Account does not exist, or if you do not have permission to access this account.

Get Security Groups

This API will return all the groups of account which are not standalone. There is a separate API for getting standalone groups.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/groups
```

Parameters: You can include:

Attribute	Description	Type
namesOnly	If set to **true**, it will only populate the name field in the security groups. Default is set to **false**.	Boolean

Body: None

Response: If the task completes, Status Code 200 OK is returned with a *Security Group List DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have permission to get the account info, but do not have permission to view the security groups."
- If the Account does not exist, or you do not have permission to access this account.

JSON Example: Get Security Groups Response

```
{
    "groups": [
        {
            "name": "Guinea-pig-group",
            "entries": [
            "exceptions":
                {
                    "type": "MX",
                    "permission": "READ"
                },
                {
                    "type": "ZONE",
                    "permission": "READ"
                },
                {
                    "type": "A",
                    "permission": "READ"
                },
                {
                    "type": "ACCOUNT_PREFERENCES",
                    "permission": "READ"
                },
                {
                    "type": "HINFO",
                    "permission": "READ"
                },
                {
                    "type": "REPORTS",
                    "permission": "READ"
                }
            ]
            "usersCount": 5
}
```

Get Users in a Security Group

This call returns a list of all of the Users assigned to a specified groupName, which are sorted by user name in ascending order.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/groups/{groupName}/users
```

Parameters:

Attribute	Description	Type
namesOnly	If set to **true**, it will only populate the name field in the *User DTO*. Default is set to **false**.	Boolean

Body: None

Response: If task completes, Status Code 200 OK is returned with a *User List DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to view security groups.
- If the Group does not exist, the Account does not exist, or if you do not have permission to access this account.

Get Standalone Security Groups

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/standalone
```

Parameters:

Attribute	Description	Type
namesOnly	If set to **true**, it will only populate the name field in the security group DTOs. Default is set to **false**.	Boolean

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Security Group List DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to view security groups.

- If the User does not exist, if the {accountName} does not exist, or if you do not have permission to access this account.

Get Settings for a Standalone Security Groups

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/standalone/{userName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Security Group DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to view security groups.
- If the User does not exist, if the {accountName} does not exist, or if you do not have permission to access this account.

Get All Exceptions for an Object

This function returns the exceptions for a specified object across all Security Groups. To make changes, you will need to edit the settings for the security group.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/exceptions/{type}/{obj
ectName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Security Exception List DTO* in the body content.

Errors: An error code is returned under the following conditions:

- If you have access to the account, but you do not have permission to view security groups.
- If the user does not exist, if {accountName} does not exist, or if you do not have permission to access the account.

Update a Security Group

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
```

Parameters: None

Body: Must include a *Security Group DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the group does not exist, the {accountName} does not exist, or if you do not have permission to access this account.

JSON Example: Update Security Group without Exceptions

```
{
  "entries": [
                    {
                     "type": "RESOURCE_RECORDS",
                     "permission": "DELETE"
                     }
                ]
}
```

JSON Example: Update Security Group with Exceptions

```
{
  "entries": [
                    {
                     "type": "ACCOUNT",
                     "permission": "DELETE"
                     }
                ],
  "exceptions":
                {
                   "exceptions":
                           [
                              {
                              "type": "ZONE",
                              "permission": "CREATE",
                              "name": "zoneName"
                              }
                           ]
                 }
}
```

Partially Update a Security Group

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
```

Parameters: None

Body: Must include a *Security Group DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the group does not exist, if the {accountName} does not exist, or if you do not have permission to access this account.

JSON Example: Partial Update Security Group without Exceptions

```
{
  "entries": [
                  {
                   "type": "ACCOUNT",
                   "permission": "CREATE"
                   },
                  {
                   "type": "RESOURCE_RECORDS",
                   "permission": "DELETE"
                   },
                  {
                   "type": "DOMAIN_SERVICES",
                   "permission": "READ"
                  },
                  {
                   "type": "ZONE",
                   "permission": "READ"
                  },
                  {
                   "type": "REPORTS",
                   "permission": "READ"
                  }
               ]
}
```

JSON Example: Partial Update Security Group with Exceptions

```
{
  "entries": [
                  {
                   "type": "ACCOUNT",
                   "permission": "DELETE"
                   }
               ],
  "exceptions":
               {
                  "exceptions":
                          [
                             {
                             "type": "ZONE",
                             "permission": "CREATE",
                             "name": "zoneName"
                             }
```

```
                        ]
                }
}
```

Update Settings for a Standalone Security Group

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/standalone/{userName}
```

Parameters: None

Body: Must include a *Security Group DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the User does not exist, if the {accountName} does not exist, or if you do not have permission to access this account.

JSON Example: Update Settings for Standalone Security Group without Exceptions

```
{
  "entries": [
                {
                 "type": "ACCOUNT",
                 "permission": "CREATE"
                 },
                {
                 "type": "RESOURCE_RECORDS",
                 "permission": "DELETE"
                 }
              ]
}
```

JSON Example: Update Settings for Standalone Security Groups with Exceptions

```
{
  "entries": [
                {
                 "type": "ACCOUNT",
                 "permission": "DELETE"
                 },
                {
                 "type": "RESOURCE_RECORDS",
                 "permission": "CREATE"
                 }
              ],
  "exceptions":
                {
                   "exceptions":
```

```
                            [
                               {
                               "type": "ZONE",
                               "permission": "CREATE",
                               "name": "zoneName"
                               }
                            ]
                }
}
```

Partially Update Settings for a Standalone Security Group

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/standalone/{userName}
```

Parameters: None

Body: Must include a *Security Group DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the User does not exist, if the {accountName} does not exist, or if you do not have permission to access this account.

JSON Example: Partial Update Settings for Standalone Security Group without Exceptions

```
{
  "entries" : [
                    {
                     "type": "RESOURCE_RECORDS",
                     "permission": "DELETE"
                     }
                ]
}
```

JSON Example: Partial Update Settings for Standalone Security Group with Exceptions

```
{
  "entries": [
                    {
                     "type": "ACCOUNT",
                     "permission": "DELETE"
                     }
                ],
  "exceptions":
                {
                   "exceptions":
                           [
                              {
```

```
                                "type": "ZONE",
                                "permission": "CREATE",
                                "name": "zoneName"
                                }
                            ]
                    }
}
```

Assign User to a Security Group

This call will perform the two following operations:

- **Move a User to different group of the same account:** If the specified user already existis in another security group (of the specified account already mentioned in request URL), this call will remove the user from their existing security group and move them to the specified security group (mentioned in the URL). If the user is a standalone user, the standalone group will be deleted.
- **Add User to the group of another account:** If the specified user does not exists in any security group (of the specified account mentioned in the request URL), this call will add the user in the specified security group of the specified account (mentioned in the URL). This will not remove the user from their existing security group.

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
/users/{userName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the User does not exist, if the Account does not exist, or if you do not have permission to access this account, or if the user does not belong to any of the accounts you have access to.
- If the account is not in an Active status.

Move User to Standalone Security Group

This call will move the user from their current security group. If the user is already a standalone user, then no action will be taken for the user.

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/standalone/{userName}
```

Parameters: None

Body: None

Response: If task completes, Status code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the User does not exist, if the Account does not exist, or if you do not have permission to access this account.
- If the account is not in an Active status.

Delete Security Group

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/groups/{groupName}
```

Parameters: None

Body: None

Response: If delete completes, Status Code 204 is returned with no body content..

Errors: An error code is returned under the following conditions:

- If you have permission to get the account info, but do not have permission to delete security groups.
- If you do not have permission to access the account.

Delete Access of a User from a Security Group

Method and URI:

```
DELETE
https://api.ultradns.com/accounts/{accountName}/groups/{groupName}/users/{use
rName}
```

Parameters: None

Body: None

Response: If task completes, Status Code 204 is returned with no body content.

Errors: An error code is returned under the following conditions:

- If you have permission to access the account, but do not have permission to modify security groups.
- If the User does not exist, if the Account does not exist, or if you do not have permission to access this account.

Security Preferences

SecurityPreferences DTO

Table 111 SecurityPreferences DTO

Attribute	Description	Type
oldPassword	The old password for the user. Mandatory only if the **password** parameter is provided in the input. Only used for setting, never returned from the server.	String.
password	The password for the user. Only used for setting, never returned from the server.	String. Password must be between 6-36 characters in length, and include at least 3 of the following: an Uppercase letter, a Lowercase letter, a Numerical digit, or a Special Character (such as: +, $, !, %). Spaces are not allowed. Additional Special Characters now supported include: @ , . / ` ~ < ? ; ' : \\ \" [ ] { } | ! # $ % ^ & * ( ) - = _ +
passwordExpiration	Maximum number of days until the password expires.	Number. You can set the value to zero so that a password change will never be required.
securityQuestion1	The ID for the security question to use.	String. (Currently a number from 1-12)
securityAnswer1	The answer to securityQuestion1.	String. securityAnswer1 length must not be greater than 36 characters.
securityQuestion2	The ID for the security question to use.	String. (Currently a number from 1-12)
securityAnswer2	The answer to securityQuestion2.	String. securityAnswer2 length must not be greater than 36 characters.
securityQuestion3	The ID for the security question to use.	String. (Currently a number from 1-12)
securityAnswer3	The answer to securityQuestion3	String. securityAnswer3 length must not be greater than 36 characters.
restricAccessIPs	IP addresses and ranges that are allowed to connect to the ingestion applications as the user.	List of *RestrictAccessIP DTO*.
inactivityTimeout	Maximum inactive duration (in minutes) after which the Java UI session will expire.	Integer.

JSON Example: Security Preferences DTO

```
{
   "oldPassword": "oldpassword",
   "password": "password",
   "passwordExpiration": "90",
```

```
   "securityQuestion1": "1",
   "securityAnswer1": "white",
   "securityQuestion2": "3"
   "securityAnswer2": "tiger",
   "securityQuestion3": "8",
   "securityAnswer3": "2008",
   "restrictAccessIPs": [
      {
         "startIP": "1.1.1.1",
         "endIP": "2.2.2.2",
         "comment": "Unit Test"
      }
   ]
   "inactivityTimeout": 15
}
```

Security Question DTO

Table 112 SecurityQuestion DTO

Attribute	Description	Type
id	The id for the security question.	String. (Currently set as a number from 1-12)
question	The text for the security question.	String.

JSON Example: SecurityQuestion DTO

```
{
      "id": "1",
      "question": "question1"
}
```

Security Question List DTO

Table 113 SecurityQuestList DTO

Attribute	Description	Type
questions	The list of the security questions.	List of *Security Question DTO* objects.

JSON Example: SecurityQuestionLIst DTO

```
{
   "questions": [
      {
      "id": "1",
      "question": "question1"
      }
   ]
}
```

RestrictIP DTO

This is for restricting the IPs that are allowed to outbound transfer primary zones.

Users will not be able to replace overlapping restrict IPs via PATCH, whereas non-overlapping restrict IPs can be added. If a user wants to add overlapping restrict IP range, then PUT (full update) should be used.

Table 114 RestrictIP DTO

Attribute	Description	Type
startIP	The starting IP address for the range. Inclusive	IPv4 Address.
endIP	The ending IP address for the range. Inclusive	IPv4 Address.
cidr	The IP range represented using CIDR. This can be used to specify a new or updated Restrict IP, but responses will always be sent as startIP/endIP pairs.	IPv4 CIDR range.
singleIP	A single IP address. This can be used to specify a new or updated Restrict IP, but responses will always be sent as startIP/endIP pairs.	IPv4 Address.
comment	A description of the range. Optional.	String.

JSON Example: RestrictIP

```
{
  "startIP": "1.2.3.4",
  "endIP": "1.2.3.4",
}
```

RestrictAccessIP DTO

This is used to control which IP addresses are allowed to connect to the ingestion applications (REST API or Java UI). It can be specified per-account or per-user. The per-account settings allow for application-specific restrictions.

Table 115 RestrictAccesIP DTO

Attribute	Description	Type
startIP	The starting IP address for the range. Inclusive	IPv4 Address.
endIP	The ending IP address for the range. Inclusive	IPv4 Address.
cidr	The IP range represented using CIDR. This can be used to specify a new or updated Restrict IP, but responses will always be sent as startIP/endIP pairs.	IPv4 CIDR range.

Attribute	Description	Type
singleIP	A single IP address. This can be used to specify a new or updated Restrict IP, but responses will always be sent as startIP/endIP pairs.	IPv4 Address.
comment	A description of the range. Optional.	String
application	Indicates whether this restriction should apply to ingestion applications, the UltraDNS Portal (UI), or the REST API application (API). Optional. If not specified, defaults to meaning all applications.	Valid values are: ▪ UI ▪ API

JSON Example: Restrict Access of IP

```
{
   "restrictAccessIPs": [
    {
      "startIP": "1.1.1.1",
      "endIP": "255.255.255.255",
      "comment": "Restrict IP.",
      "application": "UI"
    }
  ]
}
```

Get Security Questions

Method and URI:

```
GET https://api.ultradns.com/securityQuestions
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with *Security Question List DTO* and *Security Question List DTO* in the body content.

Errors: An error is returned under the following conditions:

- None

JSON Example: Get Security Questions Response

```
{
    "questions": [
        {
            "id": "1",
            "question": "What color was your first vehicle?"
        },
        {
            "id": "2",
            "question": "What year was your mother born?"
        },
        {
            "id": "3",
            "question": "What was the name of your first pet?"
        },
        {
            "id": "4",
            "question": "Where did you go on your honeymoon?"
        },
        {
            "id": "5",
            "question": "What is your high school mascot?"
         }
   ]
}
```

Get Security Preferences for a Current User

Method and URI:

```
GET https://api.ultradns.com/security
```

Parameters: None

Body: None

Response: If task completes , Status Code 200 OK is returned with *Security Question List DTO* and *Security Question List DTO* in the body content.

Errors: An error is returned under the following conditions:

- None

JSON Example: Get Security Preferences for a User Response

```
{
    "passwordExpiration": 90,
    "securityQuestion1": "1",
    "securityAnswer1": "Answer1",
    "securityQuestion2": "3",
    "securityAnswer2": "tiger",
    "securityQuestion3": "8",
    "securityAnswer3": "2008",
    "inactivityTimeout": 15
}
```

Update Security Preferences for a Current User

Method and URI:

```
PUT https://api.ultradns.com/security
```

Parameters: None

Body: Must include a *SecurityPreferences DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- Invalid data was submitted in the body.

Partially Update Security Preferences for a Current user

Method and URI:

```
PATCH https://api.ultradns.com/security
```

Parameters: None

Body: Must include a *SecurityPreferences DTO*.

Response: If task completes , Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- Invalid data was submitted in the body.

System Preferences DTO

Table 116 System Preferences DTO

Attribute	Description	Type
autoCreatePTR	**true** if PTR records should be auto-created when A records are created, **false** otherwise.	Boolean.

JSON Example: System Preferences DTO

```
{
      "autoCreatePTR": True
}
```

Get System Preferences for a Current User

Method and URI:

```
GET https://api.ultradns.com/prefs
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with *System Preferences DTO* in the body content.

```
   "restrictAccessIPs": [
    {
      "startIP": "1.1.1.1",
      "endIP": "255.255.255.255",
      "comment": "Restrict IP.",
      "application": "UI"
    }
  ]
}
```

Errors: An erorr is returned under the following conditions:

- None

Update System Preferences for a Current User

Method and URI:

```
PUT https://api.ultradns.com/prefs
```

Parameters: None

Body: Must include a *System Preferences DTO*.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.

Partially Update System Preferences for a Current User

Method and URI:

```
PATCH https://api.ultradns.com/prefs
```

Parameters: None

Body: Must include a *System Preferences DTO.*

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body.

Account Settings API

Using the Account Setting API, users can obtain and manage information for various account level settings. The following details contain the account level settings that can be managed using the available Account Setting APIs.

Table 117 Supported Account Settings

Setting Name	Description	DTO
ZONE_TRANSFER_NOTIFICATION	When there are Zone Transfer updates or failures, this setting allows you to customize the recipient(s) that should be notified, how they should be notified, and the threshold limit at which the notifications should be sent.	*NotificationSetting DTO*
DDOS_NOTIFICATION	When there are DDOS notifications, this section allows you to customize the recipient(s) that should be notified, and how they should be notified.	*NotificationSetting DTO*

GET Master Catalog for all Account Level Settings

Using this API, users can get list of all supported account level settings in the system.

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/settings/
```

Parameters: None

Body: None

Response: If task completes, Status Code 200 OK is returned with an *AccountSettingsCatalogItem List DTO* in the response body.

Errors: An error is returned under the following conditions:

- You do not have permission to read account settings for the specified {accountName}

Create Account Setting for an account

Method and URI:

```
POST https://api.ultradns.com/accounts/{accountName}/settings/{settingName}
```

Parameters: settingName can be one of from above table in path param

Body: *NotificationSetting DTO* as per path param specified in the endpoint URL

Since this is a POST function, all account settings and all the fields in the DTOs must be specified, otherwise the user will get validations error.

Response: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body around any validation error
- Given {settingName} already exists in the system
- You do not have permission to configure account settings for the specified {accountName}

Update Account Setting for an account

Method and URI:

```
PUT https://api.ultradns.com/accounts/{accountName}/settings/{settingName}
```

Parameters: settingName can be one of from above table in path param

Body: *NotificationSetting DTO*

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body around any validation error
- Given {settingName} does not exists in the system
- You do not have permission to configure account settings for the specified {accountName}

Partially Update Account Setting for an account

Method and URI:

```
PATCH https://api.ultradns.com/accounts/{accountName}/settings/{settingName}
```

Parameters: settingName can be one of from above table in path param

Body: Since this is a PATCH function, only the specified *NotificationSetting DTO* being updated need to be included in the list. All settings in the *NotificationSetting DTO* not included in the list will retain their current state.

Response: If task completes, Status Code 200 OK is returned with an appropriate status message in the response body.

Errors: An error is returned under the following conditions:

- Invalid data was submitted in the body around any validation error
- Given {settingName} does not exists in the system

- You do not have permission to configure account settings for the specified {accountName}

GET Account Setting for an account

Method and URI:

```
GET https://api.ultradns.com/accounts/{accountName}/settings/{settingName}
```

Parameters: settingName can be one of from above table in path param

Body: None

Response: If task completes, Status Code 200 OK is returned with a *NotificationSetting DTO* in the response body.

Errors: An error is returned under the following conditions:

- Given {settingName} does not exists in the system
- You do not have permission to read account settings for the specified {accountName}

Delete Account Setting for an account

Method and URI:

```
DELETE https://api.ultradns.com/accounts/{accountName}/settings/{settingName}
```

Parameters: settingName can be one of from above path param

Body: None.

Response: If delete completes, Status Code 204 is returned with no content in the response body.

Errors: An error is returned under the following conditions:

- You do not have permission to delete account settings for the specified {accountName}.
- Given {settingName} does not exists in the system.

Table 118 AccountSettingsCatalogItem DTO

Attribute	Description	Type
displayName	The display name of account level setting	One of the following account level setting as mentioned in Supported Account Settings table String type.
uri	The uri for performing the operation of account level setting	String type.

A request is composed of a list of *AccountSettingsCatalogItem DTO*. This is the structure used to request via API calls.

Table 119 AccountSettingsCatalogItem List DTO

Attribute	Description	Type
accountSettingsCatalogItems	The specified account settings for the account	List of *AccountSettingsCatalogItem DTO*.

JSON Example: AccountSettingList DTO

```
"accountSettingsCatalogItems" : [
    {
        "displayName" : "Zone Transfer Notifications"
        "uri" : "/acccount/{accountName}/settings/ZONE_TRANSFER_NOTIFICATION"
    },
    {
        "displayName" : "DDOS Notifications"
        "uri"  : "/acccount/{accountName}/settings/DDOS_NOTIFICATIONS"
    }
]
```

Table 120 NotificationSetting DTO

Attribute	Description	Type
threshold	The specified threshold when met user will receive configured notifications. This property is not applicable for DDOS_NOTIFICATION setting. If specified then it will be ignored.	Integer type.
emailNotification	The specified account settings for email notification	*EmailNotification DTO* type.

Table 121 EmailNotification DTO

Attribute	Description	Type
enable	Account setting for enabling/disabling Email pool level notification	Boolean type.
emails	The specified account settings for configuring email Ids for receiving Email notification	List of String

JSON Example: NotificationSetting DTO

The below example is applicable for the ZONE_TRANSFER_NOTIFICATION setting:

```
{
  "threshold": 200,
```

```
  "emailNotification": {
    "enable": true,
    "emails": [
      "test@test.com"
    ]
  }
}
```

The following example is applicable for the DDOS_NOTIFICATION setting:

```
{
  "emailNotification": {
    "enable": true,
    "emails": [
      "test@test.com"
    ]
  }
}
```

Batch API

The Batch API provides the ability to run multiple requests as a single unit. The Rest API will always process the individual requests sequentially, following the request order in the Batch API payload. The set of configuration changes contained amongst all of the requests in the batch will be committed to the UltraDNS configuration database, upon successful completion of all the requests

If any request in the batch fails, the batch request ends and all of the requests are then rolled back with an error message returned.

For instance, if there are four requests in a Batch API payload (A,B,C and D) and one of the requests fails (request C), the REST API will rollback everything contained in the Batch API request.

If your batch request contains GET calls only, using *Batch Query API* is typically more appropriate because if does not stop the execution when an individual call returns a 404 response status.

It is legal to put a single POST, PUT, PATCH, or DELETE call in a batch, but it will have the same semantics as making the single call directly.

The current size limit for a BATCH API request is 1,000 records. If the records exceed 1,000, a 4xx validation error message will be returned.

Asynchronous (Async)

Async is a new parameter that can be added to the method call for large Batch API requests that might time out. When this value is set to **true**, the Batch API will run the call as a background task, and provide an X-Task-ID that can be used to retrieve the status of the task.

The default value for async is false, so unless it is set to true, Batch API calls will run as normal.

Depending on the size of your Batch (or Batch Query) request, and on the size of your UltraDNS dataset, the Batch request can take significant time to execute. In certain cases, its execution time may exceed the REST API service idle timeout, resulting in a 504 response status.

When Batch (or Batch Query) is run in an asynchronous mode (by providing HTTP parameter async=true), it is not subjected to the REST API service idle timeout. Therefore, the asynchronous method should be preferred for the larger, more complex requests..

Create a Batch Request

Method and URI:

```
POST https://restapi.ultradns.com/v1/batch
```

Parameters: None

Body: Requires the use of a *Batch Request DTO*.

Response: If task completes, Status code 200 OK is returned for the batch call itself, and a *Batch Response DTO* in the body content.

NOTE: The Batch Response DTO contains individual status codes returned for each call included in the batch.

Errors: An error is returned under the following conditions:

- Error DTOs will be embedded in the response for invalid requests.
- Error DTO returned when invalid methods or URIs are used.

JSON Example: Batch API Request

```
[
  {
    "method": "POST",
    "uri": "/v1/zones/example.invalid/rrsets/A/foo",
    "body": {
      "ttl": 300,
      "rdata": ["1.2.3.4"]
    }
  },
  {
    "method": "POST",
    "uri": "/v1/zones/example.invalid/rrsets/A/bar",
    "body": {
      "ttl": 300,
      "rdata": ["10.20.30.40"]
    }
  },
  {
    "method": "DELETE",
    "uri": "/v1/zones/example.invalid/rrsets/A/baz"
  }
]
```

JSON Example: Batch API Response

```
[
    {
        "status": 201,
        "response": {
            "message": "SUCCESSFUL"
        }
    },
    {
        "status": 201,
        "response": {
            "message": "SUCCESSFUL"
        }
    },
```

```
    {
        "status": 204
    }
]
```

Create a Batch Request using Async

The **async** parameter, when added to the method, allows you to create a Batch request and receive a taskID while the operation executes in the background instead of receiving a 504 response status due to timeout.

Method and URI:

```
POST https://restapi.ultradns.com/v1/batch?async=true
```

Body: Requires the use of a *Batch Request DTO*.

Once your Batch request is completed, and you have used the X-Task-Id to retrieve your result, your next step is to delete the task from the REST API.

Method and URI:

```
DELETE https://restapi.ultradns.com/v1/tasks/{X-Task-Id}
```

Response: If task completes, Status Code 204 is returned with no body content.

JSON Example: Creating Batch Request for GlobalIP Groups with async=true

```
[
  {
    "method": "POST",
    "uri": "v1/accounts/teamrest/dirgroups/ip/foo",
    "body": {
      "name": "foo",
      "description": "Sample ip group for foo",
      "ips": [
        {
          "cidr": "172.0.0.0/29"
        },
        {
          "cidr": "172.0.0.8/29"
        },
        {
          "cidr": "172.0.0.16/29"
        }
      ]
    }
  },
  {
    "method": "POST",
    "uri": "v1/accounts/teamrest/dirgroups/ip/bar",
    "body": {
      "name": "bar",
      "description": "Sample ip group for bar",
      "ips": [
```

```
        {
          "cidr": "172.0.0.24/29"
        },
        {
          "cidr": "172.0.0.32/29"
        }
      ]
    }
  }
]
```

Response: Status code 202 is returned with a Pending message and an X-Task-Id in the header.

```
{
   "message": "Pending"
}
```

Get Task Status of Async Batch Request via X-Task-Id

From the **Headers** section, use the X-Task-Id to check the status of your pending Batch API call.

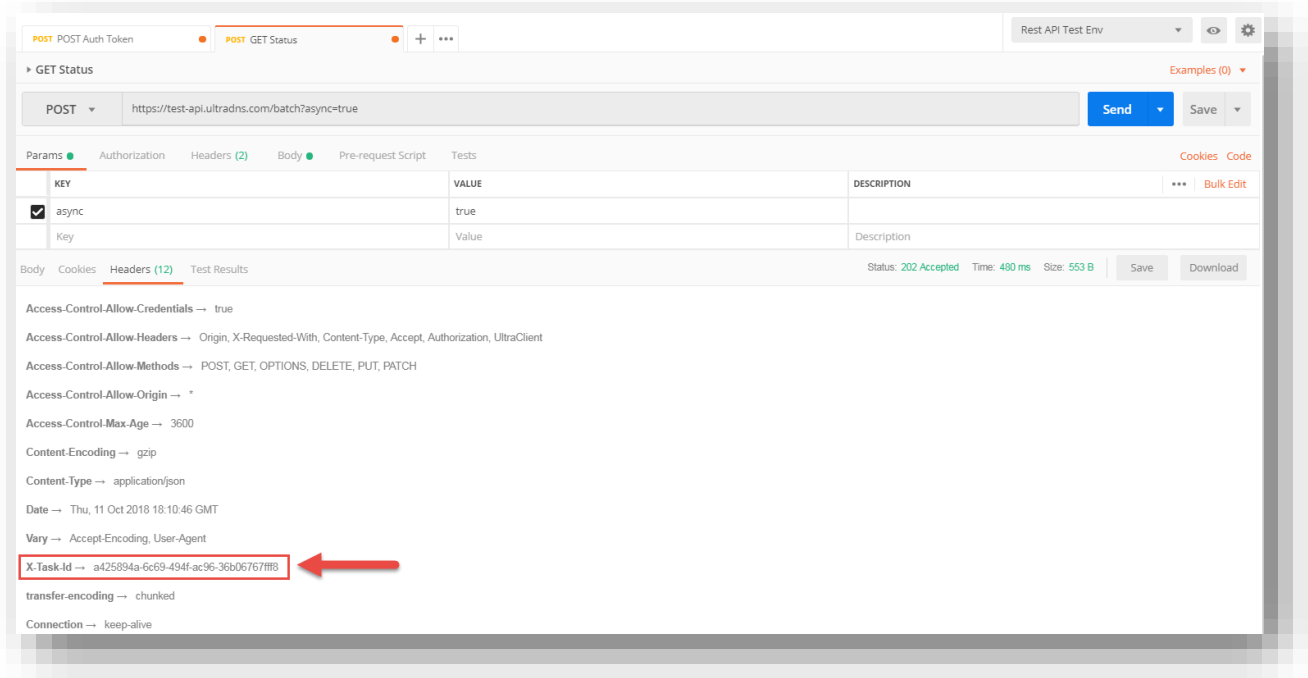

Figure 8 Async X-Task-Id in the Headers Section

Method and URI:

```
GET https://restapi.ultradns.com/v1/tasks/{X-Task-Id}
```

Parameters: None

Body: None

Response: If task completes, see the JSON example response below.

Errors: An error is returned under the following conditions:

- If the X-Task-Id is not valid.

JSON Example: Response from X-Task-Id to retrieve Batch API request status

```
{
   "taskId": "30f976cf-0d7a-48d0-99a6-002eed41868d",
   "code": "COMPLETE",
   "message": "Batch operation completed.",
   "resultUri": /tasks/30f976cf-0d7a-48d0-99a6-002eed41868d/result"
}
```

When the “code” parameter returns a “COMPLETE” message, you can use the “resultUri” to check the result of your Batch request.

In the above example, two requests were made in the Batch request, so the response will contain the status of both requests.

Get Results of Async Batch Request Using X-Task-Id

Method and URI:

```
GET https://restapi.ultradns.com/v1/tasks/{resultUri]/result
```

JSON Example: Response using resultUri to get the Batch API request

```
[
   {
      "status": 201,
      "response": {
         "message": "Successful"
      }
   }
   {
      "status": 201,
      "response": {
         "message": "Successful"
      }
   }
]
```

Delete a Batch Request X-Task-Id

Once your Batch request is completed, and you have used the X-Task-Id to retrieve your result, delete the task from the REST API.

Method and URI:

```
DELETE https://restapi.ultradns.com/v1/tasks/{X-Task-Id}
```

Parameters: None

Body: None

Response: If task completes, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- If the X-Task-Id is not valid.

Batch Request DTO

A batch request is composed of a list of Batch Request DTOs. This is the structure used to request batch changes via the API calls.

Table 122 Batch Request DTO

Field	Description	Type
method	The HTTP method that is used to submit this request. Valid values are POST, PUT, PATCH, and DELETE. GET is allowed, but is discouraged.	String.
uri	The path for the request, which includes any query parameters. Note that this is absolute, and not relative. It must start with /v1.	String.
body	Required for calls that require a body, otherwise ignored. This is the body that would be submitted for the specified method and uri.	Object.

Batch Response DTO

This is the response sent for each of the Batch Request entries. Like the Batch Request, it is composed of a list of Batch Response DTOs

Table 123 Batch Response DTO

Field	Description	Type
status	The HTTP status code returned for each call in this batch request.	Integer.
response	The body returned for each call in this batch request. For any 204 status responses (no content) that are returned for successful Delete calls, a response body will not be present.	Object.
taskId	Returned when a X-Task-ID is used to check the status of a Batch API call. (Retrieved from the Header section)	String.

Batch Query API

Similar to the Batch API function, the **Batch Query API** provides the ability to run multiple **GET** operations as a single unit. If any of the calls fail with a response status other than a 404, the request will end and an error message will be returned.

If all of the calls succeed, a success message will be returned. If a GET response returns a 404 status, the other GET requests running will return data (if there is data to provide).

Asynchronous (Async)

Async is a new parameter that can be added to the method call for large Batch API requests that might time out. When this value is set to **true**, the Batch API will run the call as a background task, and provide an X-Task-ID that can be used to retrieve the status of the task once it has finished running.

The default value for async is false, so unless it is set to true, Batch API calls will run as normal.

Depending on the size of your Batch (or Batch Query) request, and on the size of your UltraDNS dataset, the Batch request can take significant time to execute. In certain cases, its execution time may exceed the REST API service idle timeout, resulting in a 504 response status.

When Batch (or Batch Query) is run in an asynchronous mode (by providing HTTP parameter async=true), it is not subjected to the REST API service idle timeout. Therefore, the asynchronous method should be preferred for the larger, more complex requests..

The current size limit for a BATCH API request is 1,000 records. If the records exceed 1,000, a 4XX validation error message will be returned.

Create a Batch Query Request

Method and URI:

```
POST https://restapi.ultradns.com/v1/batch/query
```

Parameters: None

Body: Requires the use of a *Batch Query Request DTO*.

Response: If task completes, Status Code 200 OK is returned with a *Batch Query Response DTO* in the body content.

Errors: An error is returned under the following conditions:

- Error DTOs will be embedded in the response for invalid requests.

- Error DTO returned when invalid methods or URIs are used.

JSON Example: Batch Query Request

```
[
  {
    "method": "GET",
    "uri":"v1/zones/zone.name./rrsets"
  },
  {
    "method": "GET",
    "uri":"v1/zones/zone.name./mail/forwards"
  },
  {
    "method": "GET",
    "uri":"v1/zones/zone.name./webforwards"
  },
  {
    "method": "GET",
    "uri":"v1/zones/zone.name./mail/blocks"
  }
]
```

Create a Batch Query Request using Async

Method and URI:

```
POST https://restapi.ultradns.com/v1/batch/query?async=true
```

Parameters: None

Body: Requires the use of a *Batch Query Request DTO*.

Response: If task completes, Status Code 200 OK is returned with a *Batch Query Response DTO* in the body content.

Errors: An error is returned under the following conditions:

- Error DTOs will be embedded in the response for invalid requests.
- Error DTO returned when invalid methods or URIs are used.

JSON Example: Creating Batch Query request with async=true

```
[
  {
    "method": "GET",
    "uri": "v1/accounts/teamrest/dirgroups/ip/foo"
  },
  {
    "method": "GET",
    "uri": "v1/accounts/teamrest/dirgroups/ip/non-existing"
  },
  {
    "method": "GET",
    "uri": "v1/accounts/teamrest/dirgroups/ip/bar"
```

```
    }
]
```

Get Task Status of Async Batch Query Request via X-Task-Id

If the response returns a “Pending” message, the **Headers** section will display an X-Task-Id. Use the X-Task-Id to check the status of your pending Batch API call.

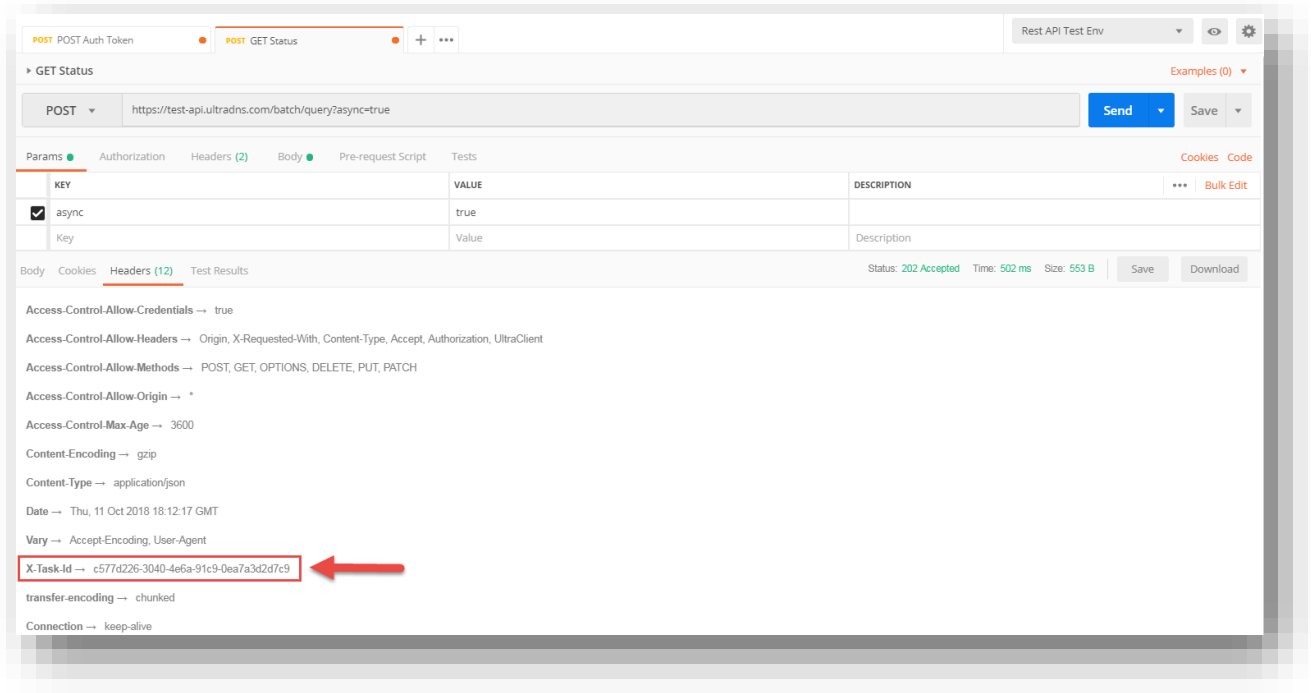

Figure 9 Batch Query Async X-Task-Id in the Headers Section

Method and URI:

```
GET https://restapi.ultradns.com/v1/tasks/{X-Task-Id}
```

JSON Example: Response using the X-Task-Id to retrieve the status of the Batch Query request

```
{
   "taskId": "30f976cf-0d7a-48d0-99a6-002eed41868d",
   "code": "COMPLETE",
   "message": "Batch operation completed.",
   "resultUri": /tasks/30f976cf-0d7a-48d0-99a6-002eed41868d/result"
}
```

When the “code” parameter returns a “Complete” message, you can then use the “resultUri” to check the result of your Batch request.

Get Results of Async Batch Query Request Using X-Task-Id

Method and URI:

```
GET  https://restapi.ultradns.com/v1/tasks/{resultUri]/result
```

JSON Example: Response using resultUri to get the Batch Query API request

```
[
   {
      "status": 201,
      "response": {
         "message": "Successful"
      }
   }
   {
      "status": 201,
      "response": {
         "message": "Successful"
      }
   }
]
```

Delete a Batch Query Request Using X-Task-Id

Once your Batch Query request is completed, and you have used the X-Task-Id to retrieve your result, your next step is to delete the task from the REST API.

Method and URI:

```
DELETE https://restapi.ultradns.com/v1/tasks/{X-Task-Id}
```

Parameters: None

Body: None

Response: If task completes, Status Code 204 is returned with no body content.

Errors: An error is returned under the following conditions:

- None

Batch Query Request DTO

A batch query request is composed of a list of Batch Query Requests DTOs. The following table provides the structure required to submit a batch query request.

Table 124 Batch Query Request DTO

Field	Description	Type
method	The HTTP method that is used to submit this request. Valid values are GET. Only GET is allowed for this API.	String.
uri	The path for the request, which includes any query parameters. Note that this is absolute, and not relative. It must start with /v1.	String.

Batch Query Response DTO

The following table provides the response that is sent for each of the Batch Query Request entries. Like the Batch request, the Batch Query response is composed of a list of Batch Response DTOs.

Table 125 Batch Query Response DTO

Field	Description	Type
status	The HTTP status code returned for this batch query request.	Integer.
response	The body returned for this batch query request. For any 204 status responses (no content) that are returned, this data will not be present.	Object.

JSON Example: Batch Query Response

```
[
  {
    "status": 200,
    "response": {
      "zoneName": "000-conversion-1.com.",
      "rrSets": [
        {
          "ownerName": "000-conversion-1.com.",
          "rrtype": "MX (15)",
          "ttl": 300,
          "rdata": [
            "10 crsmail.ultradns.net."
          ]
        },
        {
          "ownerName": "000-conversion-1.com.",
          "rrtype": "NS (2)",
          "ttl": 86400,
          "rdata": [
            "udns1.ultradns.net.",
            "udns2.ultradns.net."
          ]
        },
        {
          "ownerName": "000-conversion-1.com.",
          "rrtype": "SOA (6)",
          "ttl": 86400,
          "rdata": [
            "udns1.ultradns.net.rajender\\.aindla.neustar.biz. 2016120726
10800 3600 2592000 86400"
```

```
          ]
        }
     ]
   }
  },
  {
    "status": 404,
    "response": [
      {
        "errorCode": 70002,
        "errorMessage": "Data not found."
      }
    ]
  },
  {
    "status": 200,
    "response": {
      "queryInfo": {
        "sort": "REQUEST_TO",
        "reverse": false,
        "limit": 100
      },
      "resultInfo": {
        "totalCount": 1,
        "offset": 0,
        "returnedCount": 1
      },
      "webForwards": [
        {
         "guid": "090842F8ADF5DD6C",
         "requestTo": "owner1.000-conversion-1.com/index.html",
         "defaultRedirectTo": "http://owner2.000-conversion-1.com/index.jsp",
         "defaultForwardType": "HTTP_301_REDIRECT"
        }
      ]
    }
  }
]
```


JSON Example: Batch Query Response with async=true


```
[
    {
        "status": 200,
        "response": {
            "name": "foo",
            "description": "Sample ip group for foo",
            "ips": [
                {
                    "cidr": "172.0.0.0/29"
                },
                {
                    "cidr": "172.0.0.8/29"
                }
            ],
            "recordCount": 0
```

```
        }
    },
    {
        "status": 404,
        "response": [
            {
                "errorCode": 80001,
                "errorMessage": "Cannot find any directional group for the
given group name"
            }
        ]
    },
    {
        "status": 200,
        "response": {
            "name": "bar",
            "description": "Sample ip group for bar",
            "ips": [
                {
                    "cidr": "172.0.0.16/29"
                },
                {
                    "cidr": "172.0.0.24/29"
                },
                {
                    "cidr": "172.0.0.32/29"
                }
            ],
            "recordCount": 0
        }
    }
]
```

Reporting APIs

This document details the Neustar UltraDNS Reporting Service. This API allows you to:

- Generate Reports for various report types.
- Provide an alternative to the Report Center within the Neustar UltraDNS Managed Services Portal UI (UltraDNS Portal).

Overview

Reporting Service is a RESTful web service that provides you access to UltraDNS reports via an API interface. The reporting data retrieved is returned in response to Reporting service API calls.

Authorization

The Authorization process is outlined by the following scenarios, and is based upon the type of report that is being requested.

A. If the report requested is a Standard report, you do not need to have advanced reporting authorization set at the user account level.

B. If the report requested is an Advanced report, then the user's account will be checked for the Advanced reporting authorization before continuing.

C. If the report requested is for an account that the user belongs to, the user will be granted access to the report.

D. If the report requested is for an account that the user has access to, the user will be granted access to the report.

- Once a specific user has been authorized for a specified report, no additional authorizations will be required when retrieving information for the report, until a new report is requested.

Advanced reporting allows you to analyze time data to the minute, and location data to the client Class C address level. By comparison, the standard reporting package restricts analysis to only the hour and country level.

Projected Query Volumes

Table

Month	Day of Month	MTD Responses	Projected MTD Resp (7-day avg)	Projected MTD Resp (30-day avg)	Average TTL	Daily Responses
February	1	0				0
February	2	0				0
February	3	0				0
February	4	0				0
February	5	0				0
February	6	0				0
February	7	0				0
February	8	13			86,400	13
February	9	13				0
February	10	29			29,201	16
February	11	29				0
February	12	29				0
February	13	29				0
February	14	29				0
February	15	29				0
February	16	29				0
February	17	29				0
February	18	29				0
February	19	29				0
February	20	91			28,880	62
February	21	236	236	236		145
February	22		265	243		0
February	23		294	250		0
February	24		323	257		0
February	25		352	264		0

Rows 1 - 25

Refresh -Print -Export

Figure 10 Table view from OBIE

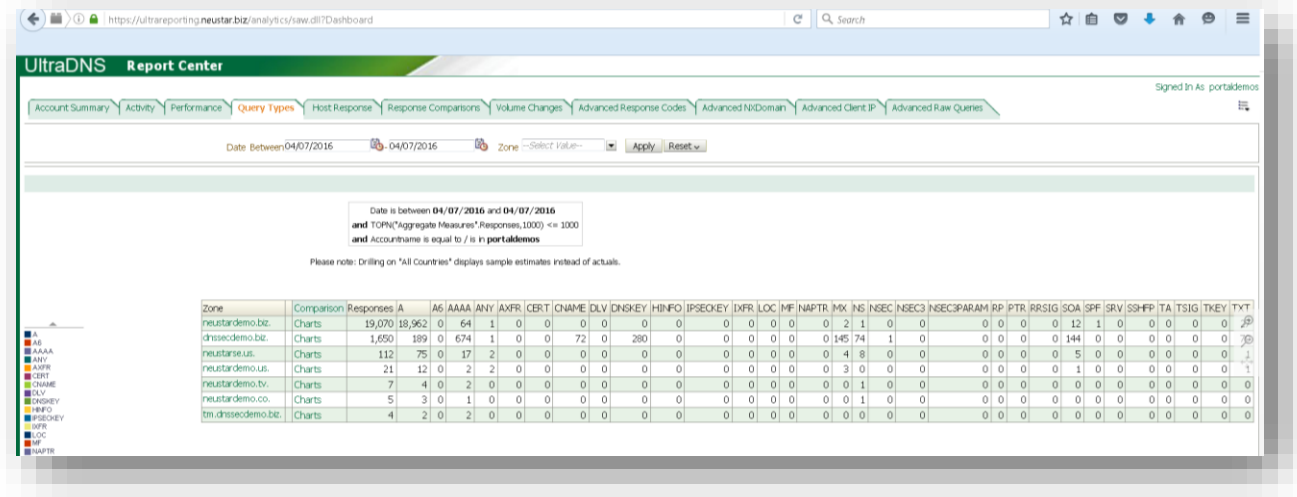

Figure 11 Ultra Activity Report - Query type

Conceptually, the output from various reports could be used to create a GUI screen or an excel table that looks like the table below.

ZoneNam	StartDate	EndDate	RspTotal	A	A6	AAAA	ANY	AXFR	CERT	CNAME	DLV	DNSKEY	HINFO	IPSECKEY	IXFR	LOC	MF	NAPTR	MX	NS	NSEC	NSEC3	NSEC3PARAM	RP	PTR	RRSIG	SOA	SI
ABC. COM	2016- 03- 01	2016- 03- 31	348238951	292481073	77	44893088	116104	123	0	306826	0	298	54	0	0	0	0	67	3935281	2086307	0	0	0	0	6954	188	31378	
DEF. COM	2016- 03- 01	2016- 03- 31	115712060	95049459	44	18352596	16870	116	0	140239	0	188	41	0	0	1	0	35	746981	629483	0	0	0	0	51	728	4580	
GHI . COM	2016- 03- 01	2016- 03- 31	94230268	76765557	35	4522787	56133	4	18	23509	0	58	0	0	0	0	0	506	2737595	130505	0	0	0	0	1E+05	0	2271681	
JKL. COM	2016- 03- 01	2016- 03- 31	82825820	60693588	112	16791914	74453	120	3	121745	0	192	73	0	0	7	10	1716	2221924	820268	4	0	0	6	3109	333	19301	
MNO. COM	2016- 03- 01	2016- 03- 31	57039919	48445277	0	7963116	3595	108	0	44240	0	158	0	0	0	0	0	0	1147	575875	0	0	0	0	3	3	787	
PQR. COM	2016- 03- 01	2016- 03- 31	51637992	43314028	18	4328781	37940	113	0	13025	0	312	20	0	0	0	1	48	1221160	1128249	0	0	0	0	24	161	2321	

Figure 12 Activity Report from Advanced Reporting

Available Reports

Table 126 Reporting Service - Available Reports

Report Name	Description	Report Type	Output Format(s)
Projected Query Volume (PQV) Report	The Projected Query Volumes report provides a snapshot of projected monthly volumes based on 7 and 30 day average query amounts. The details returned by the Projected Query Volume Report for a user that has access to multiple accounts will be consolidated across all of these accounts.	**Standard**	**JSON / CSV**
Zone Query Volume (Usage)(ZQV) Report	The Zone Query Volume report provides aggregated zone query volumes for multiple zones on a monthly basis. The data can be compiled up to 13 months.	**Standard**	**JSON / CSV**
Synchronous Zone Query Volume Report	The Zone Query Volume report provides aggregated zone query volumes for multiple zones within a 7 day period.	**Standard**	**JSON**
Zone Daily Query (ZDQV) Report	The Zone Daily Query Report provides zone query daily volume usage in aggregate for up to 60 days. This information provides detailed data that may be used to support aggregate monthly totals used for	**Standard**	**JSON / CSV**

Report Name	Description	Report Type	Output Format(s)
	billing, or for the generation of a GUI with daily data.		
Host Query Volume (HQV) Report	The Host Query Volume Report provides aggregated host query volumes for one or multiple hosts within a zone. It is intended to be a "drill down" report, designed to get more granular information on a host basis from the Zone Query Volume Report.	**Standard**	**JSON / CSV**
Host Daily Query Volume (HDQV) Report	The Host Daily Query Report provides host query daily volume usage in aggregate for up to 31 days. This information provides detailed data that may be used to support aggregate monthly totals used for billing, or for the generation of a GUI with daily data.	**Standard**	**JSON/CSV**
Probe Result Summary Report	The Probe Result Summary Report provides a summary of probe results associated to traffic service pools under a given zone for a given period of time (Max of 7 days at a time).	**Standard**	**JSON**
Probe Result Details Report	The Probe Result Details Report provides specific probe results obtained while probing a given Simple monitoring or Simple Failover traffic service pool under a given zone for a given period of time (Max of 7 days at a time).	**Standard**	**JSON**
Audit Log Report	The Audit Log Report provides up to 6 months of audit logs of activities. These activities involve DNS configuration changes (like create/update/delete of zones, records, pools), login, account/user related operations etc.	**Standard**	**JSON**
Raw Query (RQR) Report	The Raw Query report includes copious data for both queries and responses. This report can provide details for query and response troubleshooting.	**Standard**	**JSON / CSV**

Report Name	Description	Report Type	Output Format(s)
Advanced Response Codes (ARC) Report	The Advanced Response Codes report shows the DNS return codes for zones. This report can indicate a trend in DNS return codes, or pinpoint where or when specific DNS return codes began occurring in responses.	**Standard**	**JSON / CSV**
Host Level Advanced Response Codes (HARC) Report	The Host Level Advanced Response Codes report shows the DNS return codes for hosts. This report can indicate a trend in DNS return codes, or pinpoint where or when specific DNS return codes began occurring in responses.	**Standard**	**JSON / CSV**
Volume Change (VC) Report	The Volume Changes report indicates the change in volume by Zone from the previous month's volume, and can also provide from the previous 3 months and 12 months as well. This report can help identify trends in volumes for a zone and pinpoint in which month volumes increased or decreased. You can then use other reports for determine a cause.	**Standard**	**JSON / CSV**
Class C Network Level Directional Response Counts (CCNDRC) Report	The Class C Network Level Directional Response Counts Report displays the number of responses sent to Class C networks.	**Standard**	**JSON / CSV**
Cient IP Directional Response Counts (CIPDRC) Report	The Client IP Directional Response Counts Report displays the number of responses sent to Client IPs.	**Standard**	**JSON / CSV**
Zone Directional Response Counts (ZDRC) Report	The Zone Directional Response Counts Report displays the number of responses sent for zones from a specified region.	**Standard**	**JSON / CSV**
Host Directional Response Counts (HDRC) Report	The Host Directional Response Counts Report displays the number of responses sent for hosts from a specified region.	**Standard**	**JSON / CSV**
Postal Code Directional Response Counts (PCDRC) Report	The Postal Code Directional Response Counts Report displays the postal code from which the DNS query originates.	**Standard**	**JSON / CSV**

Report Name	Description	Report Type	Output Format(s)
Country Code Directional Response Coutns (CCDRC) Report	The Country Code Directional Response Counts Report displays the country codes from which the DNS queries originate.	**Standard**	**JSON / CSV**
Usage Summary Report	The Usage Summary Report displays peak data statistics for an account for the last thirty-six months.	**Standard**	**JSON / CSV**
Probe Result Summary Report	The Probe Result Summary Report returns the basic Traffic Service Probe results for an account, by displaying the successes and failures of probes during a set time period.	**Standard**	**JSON**
Probe Result Details Report	The Probe Result Details Report displays the detailed results of Traffic Service Probe results for an account, by displaying the transmitted probe details.	**Standard**	**JSON**
Audit Log Report	The Audit Log Report returns the events captured in the Audit Log for an account. These can include actions taken for a Zone or Record (add / delete / update), the creation of a user, and even the tracking of Change Comments added to an operation.	**Standard**	**JSON / CSV**
Probe Result Summary v2 Report	Probe Result Summary v2 Report returns the same results as the v1 report, but with less requirements required to generate and return the report details.	**Standard**	**JSON**
Probe Result Details v2 Report	The Probe Result Details v2 Report returns the same results as the v1 report, but with less requirements required to generate and return the report details.	**Standard**	**JSON**
Failover Report	The Failover Report displays the details for failover and/or failback events that occurred for an account.	**Standard**	**JSON**

Returning Reporting Results

Given the possibility of report sizes and the speed in which the results could be returned, reports will be returned in an asynchronous fashion. Any data that is retrieved in order to produce the report will be internally stored, and then provided once the user requests the API call to return the report data via a GET call using a **Report Request ID**.

The Request ID will be distinguishable for each Report type being requested via the prefix supplied in front of the Request ID being returned. For example, Projected Query Volume report returns a "**PQV**" prefix, while Zone Query Volume will return a "**ZQV**" prefix.

The data being held for a query request will be stored and accessible for 60 calendar days from the initial request date. The Report Request ID will only be valid for the 60 days. After which, a new report request will need to be submitted, at which time a new Report Request ID will be provided to the user to obtain the newly requested information.

Reporter Service Report Properties

Table 127 Reporter Service Report Properties

Report Name	Max Limit	Default Limit	Max Date Range	Max Number of Days in Past Allowed
Probe Summary Report	2000	25	7 Days	6 Months (185 Days)
Probe Details Report	2000	25	7 Days	6 Months (185 Days)
Projected Query Volume Report	**NA**	**NA**	**NA**	**NA**
Zone Query Volume Report	100,000	1,000	13 Months	13 Months
Synchronous Zone Query Volume Report	100,000	1,000	7 days	7 days
Zone Daily Query Volume Report	**NA**	**NA**	62 Days	13 Months
Host Query Volume Report	100,000	1,000	7 days	90 Days

Report Name	Max Limit	Default Limit	Max Date Range	Max Number of Days in Past Allowed
Host Daily Query Volume Report	**NA**	**NA**	31 Days	90 Days
Audit Report	250	50	6 Months	6 Months
Raw Query Report	10,000	1,000	90 Days	90 Days
Advanced Response Codes Report	10,000	1,000	90 Days	90 Days
Host Level Advanced Response Codes Report	10,000	1,000	90 Days	90 Days
Volume Change Report	10,000	1,000	90 Days	90 Days
Class C Network Level Directional Response Counts Report	10,000	1,000	90 Days	90 Days
Client IP Directional Response Counts Report	10,000	1,000	90 Days	90 Days
Zone Directional Response Counts Report	10,000	1,000	90 Days	90 Days
Host Directional Response	10,000	1,000	90 Days	90 Days

Report Name	Max Limit	Default Limit	Max Date Range	Max Number of Days in Past Allowed
Counts Report				
Postal Code Directional Response Counts Report	10,000	1,000	90 Days	90 Days
Country Code Directional Response Counts Report	100,000	100,000	90 Days	90 Days
Usage Summary Report	N/A	N/A	36 Months	36 Months
Probe Result Summary Report	1,000	1,000	7 Days	6 Months
Probe Result Details Report	1,000	1,000	7 Days	6 Months
Audit Log Report	250	50	6 Months	6 Months
Probe Result Summary v2 Report	1,000	1,000	7 Days	60 Days
Probe Result Details v2 Report	1,000	1,000	7 Days	60 Days
Failover Report	10,000	1,000	30 Days	90 Days

The following report figure is a representation of the fields that will be displayed in the JSON format that is returned upon using the Report ID to request the data.

URLs

Use the following base URLs for running REST API calls against the appropriate UltraDNS environment:

- Production API: https://api.ultradns.com/reports

All of the URI constructs provided in this document use the Production URL.

Calling the APIs

The UltraDNS APIs accept requests and return responses in JSON format. The default response format is JSON. A response format of .csv is available on specific reports, which are noted in this guide.

Controlling the format of the request and response is done by supplying the "Content-type" and "Accept" HTTP headers respectively, specifying **application/json** for the value in either header (or both). Keep in mind that you do not have to specify JSON for a response.

We use the Postman REST Client to provide example screenshots in this document. Postman is a freely-available REST client that allows you to save and organize frequently-used queries for later use. It can be obtained at http://www.getpostman.com/ .

Responses to API Calls

All operations return a response, and all responses have a response code (HTTP Status Code). The code number returned depends on the kind of operation you sent, (GET, POST) and the status of the operation (OK, Created).

Successful Response Codes are returned as follows:

- Status Code 200 is typically returned for a request (GET) of information and notes the call was Successful, signified by an “OK” response. If the call was a GET, you should also receive a DTO containing the information you requested.
- Status Code 201 is typically returned for a POST call and indicates that the object was Created.

If an error condition occurs, you will you will receive a 400 or 500 series (4xx or 5xx) HTTP Status Code along with an HTTP body containing a specific UltraDNS error code and a description of the error. For example:

```
[
   {
      "errorCode": 404,
      "errorMessage":"No report with the given ID was requested
                before."
   }
]
```

Response Link Headers

When using Query Parameters for certain reports, a Response Link Header may be returned to indicate that the maximum number of results exceeds the specified query parameter. In these situations, a link header indicating “next” will be provided in the response to allow you to retrieve the next set of report results. Additionally, a “previous” link will be returned as well to return to the previous set of report results.

When using the Link Header, you must re-perform the POST call for the Report along with the original body content (if any was specified) along with your new query parameters. Each subsequent request will need to be made as a POST first before retrieving your report details.

Projected Query Volume Report

Requesting Projected Query Volume Report

Method and URI:

```
POST https://api.ultradns.com/reports/dns_resolution/projected_query_volume
```

Parameters: None

Body: Must contain a *Projected Query Volume Sort DTO* and *Projected Query Volume Sortable Columns*.

Projected Query Volume Report DTO

Table 128 Projected Query Volume Sort DTO

Field	Description	Type
sortFields	Contains a map of sortable columns and sort directions.	JSON
accountName	The account name for which the report is being run against.	String.

Table 129 Projected Query Volume Sortable Columns

Sortable Column	Description	Sort Direction
month	Current Month.	ASC (Ascending) or DESC (Descending)
currentDay	Day of the Month	ASC or DESC
rspMtd	MTD Responses.	ASC or DESC
rspMtd7dAvg	Projected MTD Responses (7 day Average).	ASC or DESC
rspMtd30dAvg	Projected MTD Responses (30 day Average).	ASC or DESC
ttlAvg	Average TTL.	ASC or DESC
rspDaily	Daily Responses.	ASC or DESC

For a list of field definitions, please refer to *Table 131 Projected Query Volumes Report DTOs*.

JSON Example: Projected Query Volume Report with Sort Columns

```
{
   "projectedQueryVolume" : {
      "accountName" : "abc"
                },
{
   "sortFields":
      {
      "rspDaily": "ASC",
      "ttlAvg": "DESC",
      "rspMtd": "ASC"
      }
}
```

Responses: If task completes, Status Code 201 is returned with a *Report Request ID DTO* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."

Report Request ID DTO

The requestID is a randomly generated ID of letters and numbers sent to the user after the successful request for a report.

Table 130 ReportRequest DTO

Field	Description	Type
requestID	The requestID that is provided to the user once a request for a report has been made. ▪ For the Projected Query Volumes report, the requestID will have the following prefix: **PQV**.	String.

JSON Example: Request ID return

```
Status 201 Created
{
  "requestId": "PQV-d5a4c7ce"
}
```

Retrieving Projected Query Volumes Reports

Method and URI:

```
GET https://api.ultradns.com/requests/{requestId}
```

Parameters: *Report Request ID DTO*

Body: None

Responses: If task completes, Status Code 200 OK is returned with *Projected Query Volumes Report DTOs* in the response body. Each value is comma-separated.

Projected Query Volume Report Output DTO

Table 131 Projected Query Volumes Report DTOs

Response Body	Description	Type
year	Current year.	Integer
month	Current Month.	Month
currentDay	Day of the Month	Short
rspMtd	MTD Responses.	Long
rspMtd7dAvg	Projected MTD Responses (7 day Average).	Long
rspMtd30dAvg	Projected MTD Responses (30 day Average).	Long
ttlAvg	Average TTL.	Long
rspDaily	Daily Responses.	Long

Errors: An error code is returned under the following conditions:

- Error Code 401 – “Unauthorized. Token not found, expired or invalid.”
- Error Code 404 – “No report with the given ID was requested before.”

The Projected Query Volume Report may not be supplied immediately after providing the requestId. A status message will be returned, notifying you that the Report has not yet completed, and to try the request again at a later time.

JSON Example: Projected Query Volume Report “Pending” response

```
{
"errorCode": 410004,
"errorMessage": "Report is not ready. Please try again later."
}
```

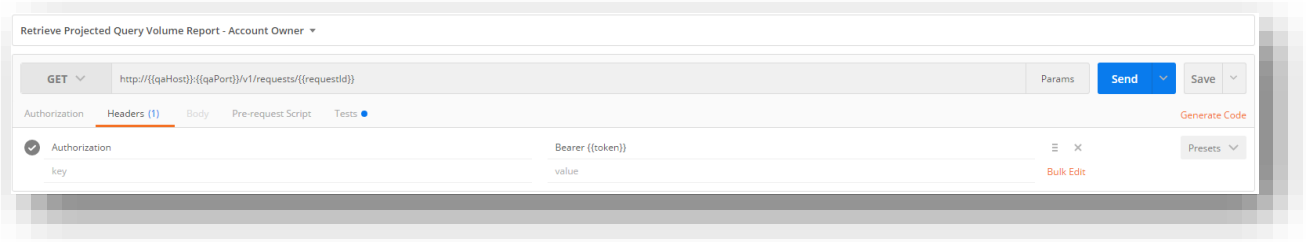

Figure 13 Projected Query Volumes Report in JSON format

JSON Example: Projected Query Volume Report return in JSON format


```
[
  {
    "year": 2016
    "month": "MARCH",
    "currentDay": 1,
    "rspMtd": 0,
    "rspMtd7dAvg": 29,
    "rspMtd30dAvg": 35,
    "ttlAvg": null,
    "rspDaily": 10,
  },
  {
    "year": 2016
    "month": "MARCH",
    "currentDay": 2,
    "rspMtd": 0,
    "rspMtd7dAvg": 34,
    "rspMtd30dAvg": 25,
    "ttlAvg": null,
    "rspDaily": 60,
  },
  {
    "year": 2016
    "month": "MARCH",
    "currentDay": 3,
    "rspMtd": 0,
    "rspMtd7dAvg": 71,
    "rspMtd30dAvg": 11,
    "ttlAvg": null,
    "rspDaily": 46,
  },
  {
    "year": 2016
    "month": "MARCH",
    "currentDay": 4,
    "rspMtd": 0,
    "rspMtd7dAvg": 32,
    "rspMtd30dAvg": 93,
    "ttlAvg": null,
    "rspDaily": 85,
  },
  {
```

```
    "year": 2016
    "month": "MARCH",
    "currentDay": 5,
    "rspMtd": 0,
    "rspMtd7dAvg": 56,
    "rspMtd30dAvg": 47,
    "ttlAvg": null,
    "rspDaily": 20,
  },
  {
```


The Projected Query Volume Report can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

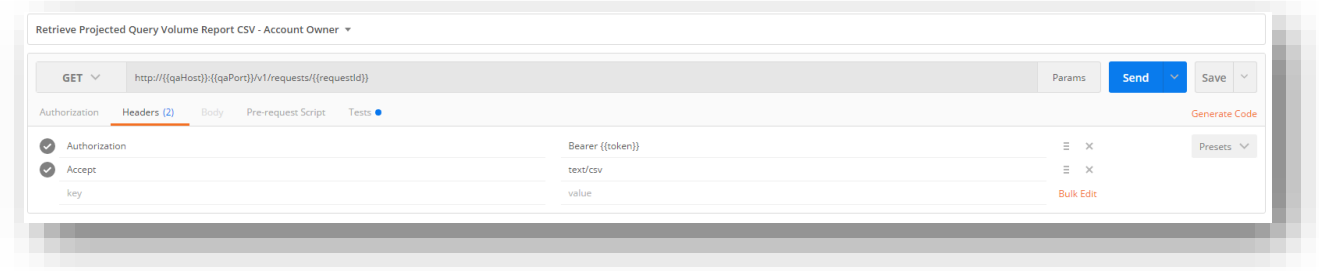

Figure 14 Projected Query Volumes Report in .CSV format

CSV Example: Projected Query Volume Report return in .CSV format


```
Month , Day Of Current Month , MTD Responses , Projected MTD Resp(7-day avg) ,
Projected MTD Resp(30-day avg) , Average TTL , Daily Responses

MARCH,1,10,29,35,30,10
MARCH,2,21,28,31,26,11
MARCH,3,33,27,39,35,12
MARCH,4,46,26,32,43,13
MARCH,5,65,36,42,1,84
MARCH,6,67,32,68,4,13
MARCH,7,51,17,3,4,24
MARCH,8,61,97,42,5,87
MARCH,9,45,7,36,2,37
MARCH,10,12,87,38,3,23
MARCH,11,82,4,19,2,59
MARCH,12,9,84,67,9,84
MARCH,13,10,31,90,2,81
MARCH,14,4,49,28,6,63
MARCH,15,2,31,57,2,72
MARCH,16,70,21,28,3,73
MARCH,17,72,59,20,22,41
MARCH,18,58,94,72,1,62
MARCH,19,72,16,12,4,43
MARCH,20,92,30,17,2,33
```

```
MARCH,21,46,80,14,20,81
MARCH,22,76,93,65,9,1
MARCH,23,59,13,88,3,58
MARCH,24,70,85,31,62,2
MARCH,25,14,87,49,20,70
MARCH,26,23,26,50,1,66
MARCH,27,84,60,88,2,31
MARCH,28,72,58,91,2,26
MARCH,29,64,52,85,5,45
MARCH,30,37,23,71,2,12
MARCH,31,59,65,21,11,84
```

Zone Query Volume Report

The purpose of the **Zone Query Volume Report** is to provide aggregated zone query volumes for multiple zones on a monthly basis, or for a set period of time. The maximum time frame between the start and end date cannot exceed 13 months.

Zone Query Volume Report DTOs

Requesting Zone Query Volume Report

Method and URI:

```
POST https://api.ultradns.com/reports/dns_resolution/query_volume/
zone?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Table 132 Zone Query Volume Query Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specific to **limit** and **SortOrder**. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for first results to be retrieved.	Integer, Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The limit will be the integer value that specifies the maximum number of results to be retrieved in response.	Integer, Optional.

Body: Must contain *Zone Query Volume DTO* and *Zone Query Volume Sort DTO*.

Zone Query Volume DTO

Table 133 Zone Query Volume DTO

Field	Description	Type
zoneName	▪ The name of the zone to be returned. ▪ The zoneName may be wildcarded with an "*" (asterisk). ▪ If not supplied, defaults to "All" zones in the account. ▪ Zone name with and without a DOT(.) at the end are supported.	String. Optional.

Field	Description	Type
accountName	The name of the account	String, Optional.
startDate	startDate **(YYYY-MM-DD)** If not supplied, defaults to the first day of the previous calendar month. The maximum number of days between the start and end date cannot exceed 13 months. ▪ startDate cannot be a future date.	String, Optional
endDate	endDate **(YYYY-MM-DD)** If not supplied, defaults to the last day of the previous calendar month. The maximum number of days between the start and end date cannot exceed 13 months. ▪ endDate cannot be a future date.	String. Optional

Table 134 Zone Query Volume Sort DTO

Field	Description	Type
sortFields	Contains a map of sortable columns and sort directions. Defaults to sort by zoneName, endDate in ascending order.	JSON

Table 135 Zone Query Volume Sortable Columns

Sortable Column	Sort Direction
zoneName	ASC (Ascending) or DESC (Descending)
startDate	ASC or DESC
endDate	ASC or DESC
rspTotal	ASC or DESC

JSON Example: Zone Query Volume Report with Sort Columns

```
{
    "zoneQueryVolume":
    {
        "startDate":"2016-05-10",
        "endDate":"2016-06-25"
    },
    "sortFields":{
        "zoneName":"DESC",
        "startDate":"ASC",
        "endDate":"DESC",
        "rspTotal":"ASC"
```

```
    }
}
```

Responses: If task completes, Status Code 201 is returned with an appropriate status message in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."

Retrieving Zone Query Volume Reports

Method and URI:

```
GET https://api.ultradns.com/requests/{requestId}
```

Parameters: *ReportRequest DTO*

Body: None

Responses: If task completes, Status Code 200 OK is returned with a list of *Zone Query Volume Report DTOs* in the response body. Each value is comma-separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Zone Query Volume Report Output DTO

Unsigned values of long are allowed from 0 to $2^{63} - 1$ (9223372036854775807)

Table 136 Zone Query Volume Report Output DTO

Response Body	Description	Type
zoneName	The name of the zone.	String
accountName	The name of the account.	String
startDate	StartDate (YYYY-MM-DD)	String
endDate	EndDate (YYYY-MM-DD)	String
rspTotal	Count of total responses between the StartDate and the EndDate across all record types.	Long
tcpTotal	Count of total TCP responses between the StartDate and the EndDate across all record types.	Long

Response Body	Description	Type
updTotal	Count of total UDP responses between the StartDate and the EndDate across all record types.	Long
ipv4Total	Count of total IPv4 responses between the StartDate and the EndDate across all record types.	Long
ipv6Total	Count of total IPv6 responses between the StartDate and the EndDate across all record types.	Long
ipv4tcpTotal	Count of total IPv4 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv4udpTotal	Count of total IPv4 UDP responses between the StartDate and the EndDate across all record types.	Long
ipv6tcpTotal	Count of total IPv6 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv6udpTotal	Count of total IPv6 UDP responses between the StartDate and the EndDate across all record types.	Long
recordA	Count of A records.	Long
recordA6	Count of A6 records.	Long
recordAAAA	Count of AAAA records.	Long
recordAny	Count of ANY records.	Long
recordAxfr	Count of AXFR records.	Long
recordCert	Count of CERT records.	Long
recordCname	Count of CNAME records.	Long
recordDlv	Count of DLV records.	Long
recordDnskey	Count of DNSKEY records.	Long
recordHinfo	Count of HINFO records.	Long
recordIpseckey	Count of IPSECKEY records.	Long
recordIxfr	Count of IXFR records.	Long
recordLoc	Count of LOC records.	Long
recordMf	Count of MF records.	Long

Response Body	Description	Type
recordNaptr	Count of NAPTR records.	Long
recordMx	Count of MX records.	Long
recordNs	Count of NS records.	Long
recordNsec	Count of NSEC records.	Long
recordNsec3	Count of NSEC3 records.	Long
recordNsec3Param	Count of NSEC3PARAM records.	Long
recordRp	Count of RP records.	Long
recordPtr	Count of PTR records.	Long
recordRrsig	Count of RRSIG records.	Long
recordSoa	Count of SOA records.	Long
recordSpf	Count of SPF records.	Long
recordSrv	Count of SRV records.	Long
recordSshfp	Count of SSHFP records.	Long
recordTa	Count of TA records.	Long
recordTsig	Count of TSIG records.	Long
recordTkey	Count of TKEY records.	Long
recordTxt	Count of TXT records.	Long

Response Link Headers

Table 137 Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** **POST** </v1/reports/dns_resolution/query_volume/zone?offset=8&limit=10>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/query_volume/zone?offset=0&limit=10>; rel="previous"

Field	Description
	When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Zone Query Volume Report without zoneName

The Zone Query Volume Report return in JSON format when zoneName is not included in the request (Defaults to all zones in the account) and multiple zones are returned.


```
[
    {
      "zoneName": "testzone9.com.",
      "accountName": "account0",
      "startDate": "2016-05-10",
      "endDate": "2016-05-28",
      "rspTotal": 12,
      "tcpTotal": 13,
      "udpTotal": 13,
      "ipv4Total": 15,
      "ipv6Total": 11,
      "ipv4tcpTotal": 8,
      "ipv4udpTotal": 6,
      "ipv6tcpTotal": 3,
      "ipv6udpTotal": 5,
      "recordA": 19,
      "recordA6": 0,
      "recordAAAA": 15,
      "recordAny": 0,
      "recordAxfr": 0,
      "recordCert": 0,
      "recordCname": 0,
      "recordDlv": 0,
      "recordDnskey": 0,
      "recordHinfo": 0,
      "recordIpseckey": 0,
      "recordIxfr": 0,
      "recordLoc": 0,
      "recordMf": 1,
      "recordNaptr": 0,
      "recordMx": 13,
      "recordNs": 19,
      "recordNsec": 0,
      "recordNsec3": 0,
      "recordNsec3Param": 0,
```

```
      "recordRp": 0,
      "recordPtr": 15,
      "recordRrsig": 0,
      "recordSoa": 0,
      "recordSpf": 0,
      "recordSrv": 0,
      "recordSshfp": 0,
      "recordTa": 0,
      "recordTsig": 0,
      "recordTkey": 0,
      "recordTxt": 0
    },
    {
      "zoneName": "testzone8.com.",
      "accountName": "account0",
      "startDate": "2016-05-10",
      "endDate": "2016-05-28",
      "rspTotal": 13,
      "tcpTotal": 17,
      "udpTotal": 13,
      "ipv4Total": 13,
      "ipv6Total": 16,
      "ipv4tcpTotal": 9,
      "ipv4udpTotal": 6,
      "ipv6tcpTotal": 8,
      "ipv6udpTotal": 8,
      "recordA": 19,
      "recordA6": 0,
      "recordAAAA": 14,
      "recordAny": 0,
      "recordAxfr": 0,
      "recordCert": 0,
      "recordCname": 0,
      "recordDlv": 0,
      "recordDnskey": 0,
      "recordHinfo": 0,
      "recordIpseckey": 0,
      "recordIxfr": 0,
      "recordLoc": 0,
      "recordMf": 1,
      "recordNaptr": 0,
      "recordMx": 13,
      "recordNs": 19,
      "recordNsec": 0,
      "recordNsec3": 0,
      "recordNsec3Param": 0,
      "recordRp": 0,
      "recordPtr": 8,
      "recordRrsig": 0,
      "recordSoa": 0,
      "recordSpf": 0,
      "recordSrv": 0,
      "recordSshfp": 0,
      "recordTa": 0,
      "recordTsig": 0,
      "recordTkey": 0,
      "recordTxt": 0
```

```
    },
    {
      "zoneName": "testzone7.com.",
      "accountName": "account0",
      "startDate": "2016-05-10",
      "endDate": "2016-05-28",
      "rspTotal": 13,
      "tcpTotal": 14,
      "udpTotal": 14,
      "ipv4Total": 18,
      "ipv6Total": 15,
      "ipv4tcpTotal": 8,
      "ipv4udpTotal": 9,
      "ipv6tcpTotal": 8,
      "ipv6udpTotal": 6,
      "recordA": 19,
      "recordA6": 0,
      "recordAAAA": 11,
      "recordAny": 0,
      "recordAxfr": 0,
      "recordCert": 0,
      "recordCname": 0,
      "recordDlv": 0,
      "recordDnskey": 0,
      "recordHinfo": 0,
      "recordIpseckey": 0,
      "recordIxfr": 0,
      "recordLoc": 0,
      "recordMf": 1,
      "recordNaptr": 0,
      "recordMx": 20,
      "recordNs": 9,
      "recordNsec": 0,
      "recordNsec3": 0,
      "recordNsec3Param": 0,
      "recordRp": 0,
      "recordPtr": 20,
      "recordRrsig": 0,
      "recordSoa": 0,
      "recordSpf": 0,
      "recordSrv": 0,
      "recordSshfp": 0,
      "recordTa": 0,
      "recordTsig": 0,
      "recordTkey": 0,
      "recordTxt": 0
    },
    {
      "zoneName": "testzone6.com.",
      "accountName": "account0",
      "startDate": "2016-05-10",
      "endDate": "2016-05-28",
      "rspTotal": 11,
      "tcpTotal": 12,
      "udpTotal": 12,
      "ipv4Total": 12,
      "ipv6Total": 10,
```

```
      "ipv4tcpTotal": 9,
      "ipv4udpTotal": 6,
      "ipv6tcpTotal": 1,
      "ipv6udpTotal": 3,
      "recordA": 12,
      "recordA6": 0,
      "recordAAAA": 10,
      "recordAny": 0,
      "recordAxfr": 0,
      "recordCert": 0,
      "recordCname": 0,
      "recordDlv": 0,
      "recordDnskey": 0,
      "recordHinfo": 0,
      "recordIpseckey": 0,
      "recordIxfr": 0,
      "recordLoc": 0,
      "recordMf": 4,
      "recordNaptr": 0,
      "recordMx": 14,
      "recordNs": 11,
      "recordNsec": 0,
      "recordNsec3": 0,
      "recordNsec3Param": 0,
      "recordRp": 0,
      "recordPtr": 14,
      "recordRrsig": 0,
      "recordSoa": 0,
      "recordSpf": 0,
      "recordSrv": 0,
      "recordSshfp": 0,
      "recordTa": 0,
      "recordTsig": 0,
      "recordTkey": 0,
      "recordTxt": 0
    },
    {
      "zoneName": "testzone5.com.",
      "accountName": "account0",
      "startDate": "2016-05-10",
      "endDate": "2016-05-28",
      "rspTotal": 13,
      "tcpTotal": 11,
      "udpTotal": 12,
      "ipv4Total": 13,
      "ipv6Total": 12,
      "ipv4tcpTotal": 5,
      "ipv4udpTotal": 7,
      "ipv6tcpTotal": 3,
      "ipv6udpTotal": 6,
      "recordA": 15,
      "recordA6": 0,
      "recordAAAA": 14,
      "recordAny": 0,
      "recordAxfr": 0,
      "recordCert": 0,
      "recordCname": 0,
```

```
      "recordDlv": 0,
      "recordDnskey": 0,
      "recordHinfo": 0,
      "recordIpseckey": 0,
      "recordIxfr": 0,
      "recordLoc": 0,
      "recordMf": 2,
      "recordNaptr": 0,
      "recordMx": 15,
      "recordNs": 13,
      "recordNsec": 0,
      "recordNsec3": 0,
      "recordNsec3Param": 0,
      "recordRp": 0,
      "recordPtr": 16,
      "recordRrsig": 0,
      "recordSoa": 0,
      "recordSpf": 0,
      "recordSrv": 0,
      "recordSshfp": 0,
      "recordTa": 0,
      "recordTsig": 0,
      "recordTkey": 0,
      "recordTxt": 0
    }
]
```


The Zone Query Volume Report can be returned in a .CSV format. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

.CSV Example: Zone Query Volume Report in .CSV format with no zoneName

Zone Query Volume Report return in .CSV format when zoneName is not included in the request (Defaults to all zones in the account) and multiple zones are returned.


```
zoneName, accountName, startDate, endDate, rspTotal, tcpTotal, udpTotal,
ipv4Total, ipv6Total, ipv4tcpTotal, ipv4udpTotal, ipv6tcpTotal, ipv6udpTotal,
recordA, recordA6, recordAAAA, recordAny, recordAxfr, recordCert,
recordCname, recordDlv, recordDnskey, recordHinfo, recordIpseckey,
recordIxfr, recordLoc, recordMf, recordNaptr, recordMx, recordNs, recordNsec,
recordNsec3, recordNsec3Param, recordRp, recordPtr, recordRrsig, recordSoa,
recordSpf, recordSrv, recordSshfp, recordTa, recordTsig, recordTkey,
recordTxt
testzone9.com.,account0,2016-05-10,2016-05-
28,12,13,13,15,11,8,6,3,5,19,0,15,0,0,0,0,0,0,0,0,0,0,1,0,13,19,0,0,0,0,15,0,
0,0,0,0,0,0,0,0
testzone8.com.,account0,2016-05-10,2016-05-
28,13,17,13,13,16,9,6,8,8,19,0,14,0,0,0,0,0,0,0,0,0,0,1,0,13,19,0,0,0,0,8,0,0
,0,0,0,0,0,0,0
testzone7.com.,account0,2016-05-10,2016-05-
28,13,14,14,18,15,8,9,8,6,19,0,11,0,0,0,0,0,0,0,0,0,0,1,0,20,9,0,0,0,0,20,0,0
,0,0,0,0,0,0,0
```

```
testzone6.com.,account0,2016-05-10,2016-05-
28,11,12,12,12,10,9,6,1,3,12,0,10,0,0,0,0,0,0,0,0,0,0,4,0,14,11,0,0,0,0,14,0,
0,0,0,0,0,0,0,0
testzone5.com.,account0,2016-05-10,2016-05-
28,13,11,12,13,12,5,7,3,6,15,0,14,0,0,0,0,0,0,0,0,0,0,2,0,15,13,0,0,0,0,16,0,
0,0,0,0,0,0,0,0
testzone4.com.,account0,2016-05-10,2016-05-
28,12,15,16,11,14,4,7,6,6,11,0,15,0,0,0,0,0,0,0,0,0,0,1,0,15,14,0,0,0,0,14,0,
0,0,0,0,0,0,0,0
```

When attempting to perform a GET for the Link Header details for the next/previous URL, a POST must first be performed to retrieve the next/previous URL of the requestId for the next set of records.

Synchronous Zone Query Volume Report

The purpose of the Synchronous **Zone Query Volume Report** is to provide aggregated zone query volumes for multiple zones on a monthly basis, or for a set period of time.

Key differences from the Zone Query Volume Report

- The maximum time frame between the start and end date cannot exceed 7 days.
- Synchronous in nature, so Report ID will be returned as results will be returned immediately.
- Significantly more details returned for each zone.
- No DTO required in the Body as this is a GET call.

Synchronous Zone Query Volume Report DTOs

Method and URI:

```
GET https://api.ultradns.com/reports/dns_resolution/query_volume/zone
```

Table 138 Synchronous Zone Query Volume Report Response DTO

Response Body	Description	Type
zoneName	The name of the zone.	String
accountName	The name of the account.	String
rspTotal	Count of total responses between the StartDate and the EndDate across all record types.	Long
tcpTotal	Count of total TCP responses between the StartDate and the EndDate across all record types.	Long
updTotal	Count of total UDP responses between the StartDate and the EndDate across all record types.	Long
ipv4Total	Count of total IPv4 responses between the StartDate and the EndDate across all record types.	Long
ipv6Total	Count of total IPv6 responses between the StartDate and the EndDate across all record types.	Long
ipv4tcpTotal	Count of total IPv4 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv4udpTotal	Count of total IPv4 UDP responses between the StartDate and the EndDate across all record types.	Long
ipv6tcpTotal	Count of total IPv6 TCP responses between the StartDate and the EndDate across all record types.	Long

Response Body	Description	Type
ipv6udpTotal	Count of total IPv6 UDP responses between the StartDate and the EndDate across all record types.	Long
recordA	Count of A records.	Long
recordA6	Count of A6 records.	Long
recordAAAA	Count of AAAA records.	Long
recordAny	Count of ANY records.	Long
recordAxfr	Count of AXFR records.	Long
recordCert	Count of CERT records.	Long
recordCname	Count of CNAME records.	Long
recordDlv	Count of DLV records.	Long
recordDnskey	Count of DNSKEY records.	Long
recordHinfo	Count of HINFO records.	Long
recordIpseckey	Count of IPSECKEY records.	Long
recordIxfr	Count of IXFR records.	Long
recordLoc	Count of LOC records.	Long
recordMf	Count of MF records.	Long
recordNaptr	Count of NAPTR records.	Long
recordMx	Count of MX records.	Long
recordNs	Count of NS records.	Long
recordNsec	Count of NSEC records.	Long
recordNsec3	Count of NSEC3 records.	Long
recordNsec3Param	Count of NSEC3PARAM records.	Long
recordRp	Count of RP records.	Long
recordPtr	Count of PTR records.	Long
recordRrsig	Count of RRSIG records.	Long
recordSoa	Count of SOA records.	Long

Response Body	Description	Type
recordSpf	Count of SPF records.	Long
recordSrv	Count of SRV records.	Long
recordSshfp	Count of SSHFP records.	Long
recordTa	Count of TA records.	Long
recordTsig	Count of TSIG records.	Long
recordTkey	Count of TKEY records.	Long
recordTxt	Count of TXT records.	Long
refusedCount	The total count of Queries refused.	Long
notimpCount	The total count of Not implemented results.	Long
nxdomainCount	The total count of Non-existent domains returned.	Long
servfailCount	The total count of Server failures returned.	Long
formerrCount	The total count of Format errors returned.	Long
noerrorCount	The total count of No errors returned.	Long

Parameters: Can include the following:

Parameter	Description	Type
sort	The sort column used to order the list. The valid values are: 1. zoneName 2. nxdomainCount 3. servfailCount 4. rspTotal Example: /reports/dns_resolution/query_volume/zone?sort=zoneName:ASC At a time, sorting is allowed only on single field. By default results will be sorted on zoneName in ascending order.	String.
offset	This field is optional. If not specified, initial records will always be returned specific to limit. This parameter allows pagination on the reporting records retrieved.The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for first results to be retrieved	Integer, Optional.

Parameter	Description	Type
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The limit will be the integer value that specifies the maximum number of results to be retrieved in response	Integer, Optional
filter	The query used to construct the list. Query operators are: 1. zoneName – Name of the zone 2. accountName – Name of account 3. startDate – Start Date from which the results needs to be fetched. By default it will be 1st of the previous month. It cannot be older than 13 months. 4. endDate – End Date till which the results needs to be fetched. By default it will be 7th day of previous month. It cannot be older than 13 months	String

Body: None

Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** **GET** </v1/reports/dns_resolution/query_volume/zone?offset=8&limit=10>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/query_volume/zone?offset=0&limit=10>; rel="previous"
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit.
Results	Total rows in the report response.

JSON Example: Synchronous Zone Query Volume Report Response Details

```
[
    {
        "zoneName": "testzone1.com.",
        "accountName": "account1",
        "rspTotal": 535,
        "tcpTotal": 0,
        "udpTotal": 535,
        "ipv4Total": 344,
        "ipv6Total": 191,
        "ipv4tcpTotal": 0,
```

```
        "ipv4udpTotal": 344,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 191,
        "recordA": 10,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 33,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 492,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 527,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 8
    },
    {
        "zoneName": "testzone2.com.",
        "accountName": "account1",
        "rspTotal": 493,
        "tcpTotal": 0,
        "udpTotal": 493,
        "ipv4Total": 303,
        "ipv6Total": 190,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 303,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 190,
        "recordA": 12,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
```

```
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 88,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 391,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 459,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 34
    },
    {
        "zoneName": "testzone3.com.",
        "accountName": "account1",
        "rspTotal": 431,
        "tcpTotal": 0,
        "udpTotal": 431,
        "ipv4Total": 173,
        "ipv6Total": 258,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 173,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 258,
        "recordA": 5,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
```

```
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 24,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 400,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 423,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 8
    },
    {
        "zoneName": "testzone4.com.",
        "accountName": "account1",
        "rspTotal": 524,
        "tcpTotal": 0,
        "udpTotal": 524,
        "ipv4Total": 322,
        "ipv6Total": 202,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 322,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 202,
        "recordA": 11,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 18,
        "recordNsec": 0,
```

```
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 494,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 520,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 4
    },
    {
        "zoneName": "testzone5.com.",
        "accountName": "account1",
        "rspTotal": 215,
        "tcpTotal": 0,
        "udpTotal": 215,
        "ipv4Total": 133,
        "ipv6Total": 82,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 133,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 82,
        "recordA": 0,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 1,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 213,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
```

```
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 213,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 2
    },
    {
        "zoneName": "testzone6.com.",
        "accountName": "account1",
        "rspTotal": 294,
        "tcpTotal": 0,
        "udpTotal": 294,
        "ipv4Total": 107,
        "ipv6Total": 187,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 107,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 187,
        "recordA": 11,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 30,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 251,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
```

```
        "notimpCount": 0,
        "nxdomainCount": 286,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 8
    },
    {
        "zoneName": "testzone7.com.",
        "accountName": "account1",
        "rspTotal": 659,
        "tcpTotal": 0,
        "udpTotal": 659,
        "ipv4Total": 375,
        "ipv6Total": 284,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 375,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 284,
        "recordA": 75,
        "recordA6": 0,
        "recordAAAA": 0,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 0,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 25,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 556,
        "recordRrsig": 0,
        "recordSoa": 0,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 651,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 8
    },
    {
```

```
        "zoneName": "testzone8.com.",
        "accountName": "account1",
        "rspTotal": 96263,
        "tcpTotal": 0,
        "udpTotal": 96263,
        "ipv4Total": 69678,
        "ipv6Total": 26585,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 69678,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 26585,
        "recordA": 4490,
        "recordA6": 0,
        "recordAAAA": 98,
        "recordAny": 20,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 197,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 7522,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 83494,
        "recordRrsig": 0,
        "recordSoa": 1,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 4340,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 91923
    },
    {
        "zoneName": "testzone9.com.",
        "accountName": "account1",
        "rspTotal": 24247,
        "tcpTotal": 0,
        "udpTotal": 24247,
        "ipv4Total": 17972,
        "ipv6Total": 6275,
```

```
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 17972,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 6275,
        "recordA": 1539,
        "recordA6": 0,
        "recordAAAA": 84,
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 45,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 2536,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 19924,
        "recordRrsig": 0,
        "recordSoa": 8,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 1839,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 22408
    },
    {
        "zoneName": "testzone10.com.",
        "accountName": "account1",
        "rspTotal": 21235,
        "tcpTotal": 0,
        "udpTotal": 21235,
        "ipv4Total": 15830,
        "ipv6Total": 5405,
        "ipv4tcpTotal": 0,
        "ipv4udpTotal": 15830,
        "ipv6tcpTotal": 0,
        "ipv6udpTotal": 5405,
        "recordA": 557,
        "recordA6": 0,
        "recordAAAA": 18,
```

```
        "recordAny": 0,
        "recordAxfr": 0,
        "recordCert": 0,
        "recordCname": 53,
        "recordDlv": 0,
        "recordDnskey": 0,
        "recordHinfo": 0,
        "recordIpseckey": 0,
        "recordIxfr": 0,
        "recordLoc": 0,
        "recordMf": 0,
        "recordNaptr": 0,
        "recordMx": 0,
        "recordNs": 2440,
        "recordNsec": 0,
        "recordNsec3": 0,
        "recordNsec3Param": 0,
        "recordRp": 0,
        "recordPtr": 18053,
        "recordRrsig": 0,
        "recordSoa": 3,
        "recordSpf": 0,
        "recordSrv": 0,
        "recordSshfp": 0,
        "recordTa": 0,
        "recordTsig": 0,
        "recordTkey": 0,
        "recordTxt": 0,
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 1788,
        "servfailCount": 0,
        "formerrCount": 0,
        "noerrorCount": 19447
    }
]
```

Zone Daily Query Report

The purpose of the **Zone Daily Query Report** is to provide Zone Query Daily Volume usage in an aggregate sum for up to 60 days. This information provides detailed data that may be used to support aggregate monthly totals used for billing, or for the generation of a GUI with daily data. The data provided via the REST API is similar to the following table (not the graphic) provided by the UltraDNS Advanced Reporting, Response Comparisons report in the GUI interface with the addition of query counts by DNS record type, protocol (UDP, TCP) and IP version. The path to the current GUI in Advanced Reporting is:

Zone Daily Query Volume Report DTOs

Requesting Zone Daily Query Volume Report

Method and URI:

```
POST https://api.ultradns.com/reports/dns_resolution/query_volume/zone/daily
```

Parameters: None

Body: Must contain *Zone Daily Query Volume DTO*.

Zone Daily Query Volume DTO

Table 139 Zone Daily Query Volume DTO

Field	Description	Type
zoneName	The name of the one zone to be returned. Wildcards in the zone name are not supported at this time. Zone name(s) with and without a DOT (.) at the end are also supported.	String.
accountName	The name of the account	String, Optional.
startDate	startDate **(YYYY-MM-DD)** If not supplied, defaults to the first day of the previous calendar month. The maximum number of days between the start and end date cannot exceed 60 days. ▪ startDate cannot be older than 13 months. ▪ startDate cannot be a future date.	String, Optional.
endDate	endDate **(YYYY-MM-DD)** If not supplied, defaults to the last day of the previous calendar month. The maximum number of days between the start and end date cannot exceed 60 days.	String. Optional.

Field	Description	Type
	▪ endDate cannot be a future date.	

Table 140 Zone Daily Query Volume Sort DTO

Field	Description	Type
sortFields	Contains a map of sortable columns and sort directions. Defaults to sort by date in ascending order.	JSON

Table 141 Zone Daily Query Volume Sortable Columns

Sortable Column	Sort Direction
date	ASC or DESC
rspTotal	ASC or DESC

For a list of field definitions, please refer to *Zone Daily Query Volume Report DTOs*

Responses: If task completes, Status Code 201 is returned with *Zone Daily Query Volume Report DTOs* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."

JSON Example: Zone Daily Query Volume Report with Sort Columns

```
{
    "zoneDailyQueryVolume":{
    "zoneName":"testzone1.com.",
    "startDate":"2016-04-01",
    "endDate":"2016-05-01"
    },
    "sortFields":{
        "date":"DESC",
        "rspTotal":"ASC"
    }
}
```

Retrieving Zone Daily Query Volume Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestId}
```

Parameters: Must contain a *ReportRequest DTO*

Body: None

Responses: If task completes, Status Code 200 OK is returned with a *Zone Daily Query Volume Report Output DTO* in the response body. Each value is comma separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Zone Daily Query Volume Report Output DTO

Unsigned values of **long** are allowed from 0 to $2^{63} - 1$ (9223372036854775807)

Table 142 Zone Daily Query Volume Report Output DTO

Response Body	Description	Type
zoneName	The name of the zone.	String
accountName	The name of the account.	String
date	Date (YYYY-MM-DD)	String
rspTotal	Count of total responses between the StartDate and the EndDate across all record types.	Long
tcpTotal	Count of total TCP responses between the StartDate and the EndDate across all record types.	Long
udpTotal	Count of total UDP responses between the StartDate and the EndDate across all record types.	Long
ipv4Total	Count of total IPv4 responses between the StartDate and the EndDate across all record types.	Long
ipv6Total	Count of total IPv6 responses between the StartDate and the EndDate across all record types.	Long
ipv4tcpTotal	Count of total IPv4 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv4udpTotal	Count of total IPv4 UDP responses between the StartDate and the EndDate across all record types.	Long
ipv6tcpTotal	Count of total IPv6 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv6udpTotal	Count of total IPv6 UDP responses between the StartDate and the EndDate across all record types.	Long
recordA	Count of A records.	Long

Response Body	Description	Type
recordA6	Count of A6 records.	Long
recordAAAA	Count of AAAA records.	Long
recordAny	Count of ANY records.	Long
recordAxfr	Count of AXFR records.	Long
recordCert	Count of CERT records.	Long
recordCname	Count of CNAME records.	Long
recordDlv	Count of DLV records.	Long
recordDnskey	Count of DNSKEY records.	Long
recordHinfo	Count of HINFO records.	Long
recordIpseckey	Count of IPSECKEY records.	Long
recordIxfr	Count of IXFR records.	Long
recordLoc	Count of LOC records.	Long
recordMf	Count of MF records.	Long
recordNaptr	Count of NAPTR records.	Long
recordMx	Count of MX records.	Long
recordNs	Count of NS records.	Long
recordNsec	Count of NSEC records.	Long
recordNsec3	Count of NSEC3 records.	Long
recordNsec3Param	Count of NSEC3PARAM records.	Long
recordRp	Count of RP records.	Long
recordPtr	Count of PTR records.	Long
recordRrsig	Count of RRSIG records.	Long
recordSoa	Count of SOA records.	Long
recordSpf	Count of SPF records.	Long
recordSrv	Count of SRV records.	Long

Response Body	Description	Type
recordSshfp	Count of SSHFP records.	Long
recordTa	Count of TA records.	Long
recordTsig	Count of TSIG records.	Long
recordTkey	Count of TKEY records.	Long
recordTxt	Count of TXT records.	Long

JSON Example: Zone Daily Query Volume Report with one zoneName

Zone Daily Query Volume Report return in JSON format when one zoneName is included in the request.


```
[
  {
    "zoneName": "testzone1.com.",
    "accountName": "account0",
    "date": "2016-05-01",
    "rspTotal": 0,
    "tcpTotal": 0,
    "udpTotal": 0,
    "ipv4Total": 1,
    "ipv6Total": 1,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 0,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 1,
    "recordA6": 0,
    "recordAAAA": 0,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
```

```
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "testzone1.com.",
    "accountName": "account0",
    "date": "2016-04-28",
    "rspTotal": 0,
    "tcpTotal": 0,
    "udpTotal": 0,
    "ipv4Total": 1,
    "ipv6Total": 1,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 0,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 0,
    "recordA6": 0,
    "recordAAAA": 0,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 1,
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 1,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "testzone1.com.",
    "accountName": "account0",
```

```
    "date": "2016-04-27",
    "rspTotal": 0,
    "tcpTotal": 1,
    "udpTotal": 0,
    "ipv4Total": 0,
    "ipv6Total": 1,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 0,
    "ipv6tcpTotal": 1,
    "ipv6udpTotal": 0,
    "recordA": 0,
    "recordA6": 0,
    "recordAAAA": 1,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
    "recordNs": 1,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 1,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "testzone1.com.",
    "accountName": "account0",
    "date": "2016-04-26",
    "rspTotal": 1,
    "tcpTotal": 0,
    "udpTotal": 0,
    "ipv4Total": 0,
    "ipv6Total": 1,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 0,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 1,
    "recordA6": 0,
```

```
    "recordAAAA": 1,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "testzone1.com.",
    "accountName": "account0",
    "date": "2016-04-25",
    "rspTotal": 0,
    "tcpTotal": 0,
    "udpTotal": 1,
    "ipv4Total": 0,
    "ipv6Total": 0,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 0,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 1,
    "recordA6": 0,
    "recordAAAA": 0,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
```

```
    "recordNaptr": 0,
    "recordMx": 1,
    "recordNs": 1,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
```


The Zone Daily Query Volume Report can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

.CSV Example: Zone Daily Query Volume Report with one zoneName

Zone Daily Query Volume Report return in .CSV format when a single zoneName is included in the request.


```
zoneName, accountName, date, rspTotal, tcpTotal, udpTotal, ipv4Total,
ipv6Total, ipv4tcpTotal, ipv4udpTotal, ipv6tcpTotal, ipv6udpTotal, recordA,
recordA6, recordAAAA, recordAny, recordAxfr, recordCert, recordCname,
recordDlv, recordDnskey, recordHinfo, recordIpseckey, recordIxfr, recordLoc,
recordMf, recordNaptr, recordMx, recordNs, recordNsec, recordNsec3,
recordNsec3Param, recordRp, recordPtr, recordRrsig, recordSoa, recordSpf,
recordSrv, recordSshfp, recordTa, recordTsig, recordTkey, recordTxt
testzone1.com.,account0,2016-05-
01,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
testzone1.com.,account0,2016-04-
28,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,
0,0,0
testzone1.com.,account0,2016-04-
27,0,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,
0,0,0
testzone1.com.,account0,2016-04-
26,1,0,0,0,1,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
testzone1.com.,account0,2016-04-
25,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
testzone1.com.,account0,2016-04-
24,1,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
```

```
testzone1.com.,account0,2016-04-
23,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
```

Host Query Volume Report

The purpose of the Host Query Volume Report is to provide aggregated host query volumes for one or all hosts within a zone. It is intended to be a “drill down” report, designed to get more granular information on a host basis from the Zone Query Volume Report.

The output columns are the same as the Zone Query Volume with the addition of the **Host Name**.

Host Query Volume Report DTOs

Requesting Host Query Volume Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/query_volume/host?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Table 143 Host Query Volume Report Query Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specific to limit and SortOrder. This parameter allows pagination on the report records retrieved. The offset will be the integer value that specifies the position of the first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer, Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the report records retrieved. The maximum number of results to be retrieved in a single response is 100,000 records.	Integer, Optional.

Body: Must contain a *Host Query Volume DTO*.

Host Query Volume DTO

Table 144 Host Query Volume DTO

Field	Description	Type
zoneName	▪ The name of the zone to be returned. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Required.
hostName	▪ The name of the host to be returned.	String. Optional.

Field	Description	Type
	▪ If not supplied, defaults to “All” hosts in the zone. ▪ The HostName is a FQDN.	
accountName	Name of the account.	String, Optional
startDate	StartDate **(YYYY-MM-DD)** If not supplied, defaults to the first day of the previous calendar month. The maximum number of days between the startDate and endDate cannot exceed 7 days. This is inclusive of startDate and endDate. ▪ The startDate must be before or equal to the endDate. ▪ startDate cannot be older than 90 days from the present date. ▪ startDate cannot be a future date.	String, Optional
endDate	EndDate **(YYYY-MM-DD)** If not supplied, defaults to the 7th day of the previous calendar month. The maximum number of days between the startDate and endDate cannot exceed 7 days. This is inclusive of startDate and endDate. ▪ endDate cannot be a future date.	String. Optional

Table 145 Host Query Volume Sort DTO

Field	Description	Type
sortFields	Contains a map of sortable columns and sort directions. Defaults to sort by hostName, endDate in ascending order.	JSON

Table 146 Host Query Volume Sortable Columns

Sortable Column	Sort Direction
hostName	ASC (Ascending) or DESC (Descending)
startDate	ASC or DESC
endDate	ASC or DESC
rspTotal	ASC or DESC

Responses: If task completes, Status Code 201 is returned with a *Host Query Volume DTO* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 400 – If report endDate is before startDate.
- Error Code 400 – If date provided is not in valid format i.e. "YYYY-MM-DD."
- Error Code 400 – When zone name is not provided.
- Error Code 400 – When any one of startDate and endDate is not provided.
- Error Code 400 – If startDate/endDate is a future date.
- Error Code 400 – If offset is a negative value.
- Error Code 400 – When startDate is older than 90 days.

JSON Example: Host Query Volume Report with Sort Columns

```
{
   "hostQueryVolume":
     {
      "zoneName": "abc.com.",
      "startDate": "2016-07-12",
      "endDate": "2016-07-18"
     },
     "sortFields":
      {
      "hostName": "ASC",
      "rspTotal": "DESC",
      "startDate": "ASC",
      "endDate": "ASC"
      }
}
```

Retrieving Host Query Volume Reports

Method and URI:

```
GET https://api.ultradns.com/requests/{requestId}
```

Parameters: Must contain a *ReportRequest DTO*

Responses: If task completes, Status Code 200 OK is returned with a list of *Host Query Volume Report DTO Output* in the response body.. Each value is comma-separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Host Query Volume Report Output DTO

Table 147 Host Query Volume Report DTO Output

Response Body	Description	Type
zoneName	The name of the zone.	Sting
hostName	The name of the host.	Sting
accountName	The name of the account.	String
startDate	The startDate **(YYYY-MM-DD)** time.	String
endDate	The endDate **(YYYY-MM-DD)** time.	String
rspTotal	Count of total responses between the StartDate and the EndDate across all record types.	Long
tcpTotal	Count of total TCP responses between the StartDate and the EndDate across all record types.	Long
updTotal	Count of total UDP responses between the StartDate and the EndDate across all record types.	Long
ipv4Total	Count of total IPv4 responses between the StartDate and the EndDate across all record types.	Long
ipv6Total	Count of total IPv6 responses between the StartDate and the EndDate across all record types.	Long
ipv4tcpTotal	Count of total IPv4 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv4udpTotal	Count of total IPv4 UDP responses between the StartDate and the EndDate across all record types.	Long
ipv6tcpTotal	Count of total IPv6 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv6udpTotal	Count of total IPv6 UDP responses between the StartDate and the EndDate across all record types.	Long
recordA	Count of A records.	Long
recordA6	Count of A6 records.	Long
recordAAAA	Count of AAAA records.	Long
recordAny	Count of ANY records.	Long
recordAxfr	Count of AXFR records.	Long
recordCert	Count of CERT records.	Long
recordCname	Count of CNAME records.	Long
recordDlv	Count of DLV records.	Long

Response Body	Description	Type
recordDnskey	Count of DNSKEY records.	Long
recordHinfo	Count of HINFO records.	Long
recordIpseckey	Count of IPSECKEY records.	Long
recordIxfr	Count of IXFR records.	Long
recordLoc	Count of LOC records.	Long
recordMf	Count of MF records.	Long
recordNaptr	Count of NAPTR records.	Long
recordMx	Count of MX records.	Long
recordNs	Count of NS records.	Long
recordNsec	Count of NSEC records.	Long
recordNsec3	Count of NSEC3 records.	Long
recordNsec3Param	Count of NSEC3PARAM records.	Long
recordRp	Count of RP records.	Long
recordPtr	Count of PTR records.	Long
recordRrsig	Count of RRSIG records.	Long
recordSoa	Count of SOA records.	Long
recordSpf	Count of SPF records.	Long
recordSrv	Count of SRV records.	Long
recordSshfp	Count of SSHFP records.	Long
recordTa	Count of TA records.	Long
recordTsig	Count of TSIG records.	Long
recordTkey	Count of TKEY records.	Long
recordTxt	Count of TXT records.	Long

Long is an unsigned 64-bit number and its value can be anything between 0 to $2^{63} - 1$ (9223372036854775807).

Response Link Headers

Table 148 Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/query_volume/host?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/query_volume/host?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k
Results	Total rows in the report response.

JSON Example: Host Query Volume Report return in JSON format when no hostName is included, and multiple hosts are returned.


```
[
  {
    "zoneName": "abc.com.",
    "hostName": "http.",
    "accountName": "account0",
    "startDate": "2016-07-15",
    "endDate": "2016-07-15",
    "rspTotal": 2,
    "tcpTotal": 0,
    "udpTotal": 2,
    "ipv4Total": 2,
    "ipv6Total": 0,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 2,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 2,
    "recordA6": 0,
    "recordAAAA": 0,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
```

```
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "abc.com.",
    "hostName": "ftp.",
    "accountName": "account0",
    "startDate": "2016-07-16",
    "endDate": "2016-07-16",
    "rspTotal": 1,
    "tcpTotal": 0,
    "udpTotal": 1,
    "ipv4Total": 1,
    "ipv6Total": 0,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 1,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 0,
    "recordA6": 0,
    "recordAAAA": 1,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
```

```
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  },
  {
    "zoneName": "abc.com.",
    "hostName": "voip.",
    "accountName": "account0",
    "startDate": "2016-07-16",
    "endDate": "2016-07-16",
    "rspTotal": 1,
    "tcpTotal": 0,
    "udpTotal": 1,
    "ipv4Total": 1,
    "ipv6Total": 0,
    "ipv4tcpTotal": 0,
    "ipv4udpTotal": 1,
    "ipv6tcpTotal": 0,
    "ipv6udpTotal": 0,
    "recordA": 0,
    "recordA6": 0,
    "recordAAAA": 1,
    "recordAny": 0,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 0,
    "recordDlv": 0,
    "recordDnskey": 0,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 0,
    "recordNs": 0,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 0,
    "recordRrsig": 0,
    "recordSoa": 0,
    "recordSpf": 0,
    "recordSrv": 0,
```

```
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 0
  }
]
```


The Host Query Volume Report can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/plain**.

.CSV Example: Host Query Volume Report return in .CSV format when no hostName is included and multiple hosts are returned.


```
zoneName, hostName, accountName, startDate, endDate, rspTotal, tcpTotal,
udpTotal, ipv4Total, ipv6Total, ipv4tcpTotal, ipv4udpTotal, ipv6tcpTotal,
ipv6udpTotal, recordA, recordA6, recordAAAA, recordAny, recordAxfr,
recordCert, recordCname, recordDlv, recordDnskey, recordHinfo,
recordIpseckey, recordIxfr, recordLoc, recordMf, recordNaptr, recordMx,
recordNs, recordNsec, recordNsec3, recordNsec3Param, recordRp, recordPtr,
recordRrsig, recordSoa, recordSpf, recordSrv, recordSshfp, recordTa,
recordTsig, recordTkey, recordTxt
abc.com.,http.,account0,2016-07-15,2016-07-
15,2,0,2,2,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
abc.com.,ftp.,account0,2016-07-16,2016-07-
16,1,0,1,1,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
abc.com.,voip.,account0,2016-07-16,2016-07-
16,1,0,1,1,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
```


When attempting to perform a GET for the Link Header details for the next/previous URL, a POST must first be performed to retrieve the next/previous URL of the requestId for the next set of records.

Host Daily Query Volume Report

The purpose of the Host Daily Query Report is to provide Host Query Daily Volume usage in aggregate for up to 31 days. This report is similar to the Zone Daily Query Volume report with the addition of the **Host Name** and **Data Aggregation** at the host level.

The output columns are the same as the zone daily query volume with the addition of host name. The maximum number of days between the start and end date cannot exceed 31 days.

Host Daily Query Volume Report DTOs

Requesting Host Daily Query Volume Report

Method and URI:

```
POST https://api.ultradns.com/reports/dns_resolution/query_volume/host/daily
```

Parameters: None

Body: Must contain a *Host Daily Query Volume DTO*.

Host Daily Query Volume DTO

Table 149 Host Daily Query Volume DTO

Field	Description	Type
zoneName	▪ The name of the one zone to be returned. ▪ Wildcards in the zone name are not supported at this time. ▪ Zone name(s) with and without a DOT (.) at the end are also supported.	String. Required
hostName	▪ The name of the host to be returned. ▪ Wildcards in the host name are not supported at this time. ▪ The HostName is a FQDN.	String. Required
accountName	The name of the account	String. Optional
startDate	startDate **(YYYY-MM-DD)** If not supplied, defaults to the first day of the previous calendar month. The maximum number of days between the startDate and endDate cannot exceed 31 days. This is inclusive of startDate and endDate. The startDate must be before or equal to endDate. ▪ startDate cannot be older than 90 days from the present date. ▪ startDate cannot be a future date.	String. Optional

Field	Description	Type
endDate	endDate **(YYYY-MM-DD)** If not supplied, defaults to the last day of the previous calendar month. The maximum number of days between the startDate and endDate cannot exceed 31 days. This is inclusive of startDate and endDate. ▪ endDate cannot be a future date.	String. Optional

Table 150 Host Daily Query Volume Sort DTO

Field	Description	Type
sortFields	Contains a map of sortable columns and sort directions. Defaults to sort by Date in ascending order.	JSON

Table 151 Host Daily Query Volume Sortable Columns

Sortable Column	Sort Direction
date	ASC or DESC
rspTotal	ASC or DESC

Responses: If task completes, Status Code 201 is returned with a *Host Daily Query Volume DTO* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 400 – If report endDate is before startDate.
- Error Code 400 – If date provided is not in valid format i.e. "YYYY-MM-DD."
- Error Code 400 – When zone name is not provided.
- Error Code 400 – When host name is not provided.
- Error Code 400 – When any one of startDate and endDate is not provided.
- Error Code 400 – If startDate/endDate is a future date.
- Error Code 400 – When startDate is older than 90 days.

JSON Example: Host Daily Query Volume Report with Sort Columns

```
{
   "hostDailyQueryVolume":
     {
      "zoneName": "abc.com.",
      "hostName": "www.abc.com",
```

```
      "startDate": "2016-07-01",
      "endDate": "2016-07-24"
     },
   "sortFields":
      {
      "date": "DESC",
      "rspTotal": "DESC"
      }
}
```

Retrieving Host Daily Query Volume Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestId}
```

Parameters: Must contain a *ReportRequest DTO*.

Body: None

Responses: If task completes, Status Code 200 OK is returned with a *Host Daily Query Volume Report DTO Output* in the response body. Each value is comma-separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 152 Host Daily Query Volume Report DTO Output

Response Body	Description	Type
zoneName	The name of the zone.	String
hostName	The name of host.	String.
accountName	The name of the account.	String
date	Date **(YYYY-MM-DD)**	String
rspTotal	Count of total responses between the StartDate and the EndDate across all record types.	Long
tcpTotal	Count of total TCP responses between the StartDate and the EndDate across all record types.	Long
updTotal	Count of total UDP responses between the StartDate and the EndDate across all record types.	Long
ipv4Total	Count of total IPv4 responses between the StartDate and the EndDate across all record types.	Long
ipv6Total	Count of total IPv6 responses between the StartDate and the EndDate across all record types.	Long
ipv4tcpTotal	Count of total IPv4 TCP responses between the StartDate and the EndDate across all record types.	Long

Response Body	Description	Type
ipv4udpTotal	Count of total IPv4 UDP responses between the StartDate and the EndDate across all record types.	Long
ipv6tcpTotal	Count of total IPv6 TCP responses between the StartDate and the EndDate across all record types.	Long
ipv6udpTotal	Count of total IPv6 UDP responses between the StartDate and the EndDate across all record types.	Long
recordA	Count of A records.	Long
recordA6	Count of A6 records.	Long
recordAAAA	Count of AAAA records.	Long
recordAny	Count of ANY records.	Long
recordAxfr	Count of AXFR records.	Long
recordCert	Count of CERT records.	Long
recordCname	Count of CNAME records.	Long
recordDlv	Count of DLV records.	Long
recordDnskey	Count of DNSKEY records.	Long
recordHinfo	Count of HINFO records.	Long
recordIpseckey	Count of IPSECKEY records.	Long
recordIxfr	Count of IXFR records.	Long
recordLoc	Count of LOC records.	Long
recordMf	Count of MF records.	Long
recordNaptr	Count of NAPTR records.	Long
recordMx	Count of MX records.	Long
recordNs	Count of NS records.	Long
recordNsec	Count of NSEC records.	Long
recordNsec3	Count of NSEC3 records.	Long
recordNsec3Param	Count of NSEC3PARAM records	Long
recordRp	Count of RP records.	Long
recordPtr	Count of PTR records.	Long
recordRrsig	Count of RRSIG records.	Long

Response Body	Description	Type
recordSoa	Count of SOA records.	Long
recordSpf	Count of SPF records.	Long
recordSrv	Count of SRV records.	Long
recordSshfp	Count of SSHFP records.	Long
recordTa	Count of TA records.	Long
recordTsig	Count of TSIG records.	Long
recordTkey	Count of TKEY records.	Long
recordTxt	Count of TXT records.	Long

Long is an unsigned 64-bit number and its value can be anything between 0 to $2^{63} - 1$ (9223372036854775807).

JSON Example: Host Daily Query Volume Report return in JSON format when one Host Name requested.


```
[
  {
    "zoneName": "abc.com.",
    "hostName": "www.abc.com",
    "accountName": "account0",
    "date": "2016-07-18",
    "rspTotal": 573957,
    "tcpTotal": 21909,
    "udpTotal": 552048,
    "ipv4Total": 491788,
    "ipv6Total": 82169,
    "ipv4tcpTotal": 20747,
    "ipv4udpTotal": 471041,
    "ipv6tcpTotal": 1162,
    "ipv6udpTotal": 81007,
    "recordA": 290063,
    "recordA6": 17,
    "recordAAAA": 66010,
    "recordAny": 2571,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 242,
    "recordDlv": 0,
    "recordDnskey": 4,
    "recordHinfo": 1,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
```

```
    "recordMf": 0,
    "recordNaptr": 1,
    "recordMx": 89816,
    "recordNs": 61858,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 19,
    "recordRrsig": 1,
    "recordSoa": 650,
    "recordSpf": 4142,
    "recordSrv": 1,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 58416
  },
  {
    "zoneName": "abc.com.",
    "hostName": "www.",
    "accountName": "account0",
    "date": "2016-07-17",
    "rspTotal": 364814,
    "tcpTotal": 3659,
    "udpTotal": 361155,
    "ipv4Total": 297793,
    "ipv6Total": 67021,
    "ipv4tcpTotal": 3307,
    "ipv4udpTotal": 294486,
    "ipv6tcpTotal": 352,
    "ipv6udpTotal": 66669,
    "recordA": 223775,
    "recordA6": 4,
    "recordAAAA": 57696,
    "recordAny": 1049,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 127,
    "recordDlv": 0,
    "recordDnskey": 4,
    "recordHinfo": 3,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 3,
    "recordMx": 30858,
    "recordNs": 41141,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 5,
    "recordRrsig": 0,
    "recordSoa": 579,
```

```
    "recordSpf": 721,
    "recordSrv": 3,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 8776
  },
  {
    "zoneName": "abc.com.",
    "hostName": "www.",
    "accountName": "account0",
    "date": "2016-07-16",
    "rspTotal": 391631,
    "tcpTotal": 4967,
    "udpTotal": 386664,
    "ipv4Total": 321913,
    "ipv6Total": 69718,
    "ipv4tcpTotal": 4445,
    "ipv4udpTotal": 317468,
    "ipv6tcpTotal": 522,
    "ipv6udpTotal": 69196,
    "recordA": 225050,
    "recordA6": 1,
    "recordAAAA": 62719,
    "recordAny": 1557,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 105,
    "recordDlv": 0,
    "recordDnskey": 4,
    "recordHinfo": 0,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 0,
    "recordMx": 45641,
    "recordNs": 41518,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 3,
    "recordRrsig": 0,
    "recordSoa": 564,
    "recordSpf": 1632,
    "recordSrv": 0,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 12747
  },
  {
    "zoneName": "abc.com.",
    "hostName": "www.",
```

```
    "accountName": "account0",
    "date": "2016-07-15",
    "rspTotal": 556847,
    "tcpTotal": 16951,
    "udpTotal": 539896,
    "ipv4Total": 473392,
    "ipv6Total": 83455,
    "ipv4tcpTotal": 15835,
    "ipv4udpTotal": 457557,
    "ipv6tcpTotal": 1116,
    "ipv6udpTotal": 82339,
    "recordA": 278007,
    "recordA6": 25,
    "recordAAAA": 63792,
    "recordAny": 2029,
    "recordAxfr": 0,
    "recordCert": 0,
    "recordCname": 206,
    "recordDlv": 0,
    "recordDnskey": 4,
    "recordHinfo": 2,
    "recordIpseckey": 0,
    "recordIxfr": 0,
    "recordLoc": 0,
    "recordMf": 0,
    "recordNaptr": 2,
    "recordMx": 105740,
    "recordNs": 59374,
    "recordNsec": 0,
    "recordNsec3": 0,
    "recordNsec3Param": 0,
    "recordRp": 0,
    "recordPtr": 24,
    "recordRrsig": 5,
    "recordSoa": 692,
    "recordSpf": 3319,
    "recordSrv": 2,
    "recordSshfp": 0,
    "recordTa": 0,
    "recordTsig": 0,
    "recordTkey": 0,
    "recordTxt": 43450
  }
]
```


The Host Daily Query Volume Report can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

CSV Example: Host Daily Query Volume Report when a single Host Name is requested.


```
zoneName, hostName, accountName, date, rspTotal, tcpTotal, udpTotal,
ipv4Total, ipv6Total, ipv4tcpTotal, ipv4udpTotal, ipv6tcpTotal, ipv6udpTotal,
recordA, recordA6, recordAAAA, recordAny, recordAxfr, recordCert,
recordCname, recordDlv, recordDnskey, recordHinfo, recordIpseckey,
```

```
recordIxfr, recordLoc, recordMf, recordNaptr, recordMx, recordNs, recordNsec,
recordNsec3, recordNsec3Param, recordRp, recordPtr, recordRrsig, recordSoa,
recordSpf, recordSrv, recordSshfp, recordTa, recordTsig, recordTkey,
recordTxt
abc.com.,www.,account0,2016-07-
18,573957,21909,552048,491788,82169,20747,471041,1162,81007,290063,17,66010,2
571,0,0,242,0,4,1,0,0,0,0,1,89816,61858,0,0,0,0,19,1,650,4142,1,0,0,0,0,58416
abc.com.,www.,account0,2016-07-
17,364814,3659,361155,297793,67021,3307,294486,352,66669,223775,4,57696,1049,
0,0,127,0,4,3,0,0,0,0,3,30858,41141,0,0,0,0,5,0,579,721,3,0,0,0,0,8776
abc.com.,www.,account0,2016-07-
16,391631,4967,386664,321913,69718,4445,317468,522,69196,225050,1,62719,1557,
0,0,105,0,4,0,0,0,0,0,0,45641,41518,0,0,0,0,3,0,564,1632,0,0,0,0,0,12747
```

Raw Query Sample Report

The Advanced Raw Queries report includes copious data for both queries and responses. This report can provide details for query and response troubleshooting. For more details about reports, review the Report Center User Guide found on the Support page of the UltraDNS Portal.

Requesting Raw Query Sample Report

Method and URI:

```
POST https://api.ultradns.com/reports/dns_resolution/sample/raw_queryHTTP/1.1
```

OR

```
POST
https://api.ultradns.com/reports/dns_resolution/sample/raw_query?offset={offset}&limit={limit}
```

Parameters: May contain the following:

Raw Query Sample Parameters

Table 153 Raw Query Parameters DTO

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain a *Raw Query Report DTO*.

Raw Query Report DTO

Table 154 Raw Query Sort DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
zoneName	The results of the one zone being returned. ▪ When not specified, all zones under the account will be queried for reporting. ▪ Wildcards in the zone name are currently not supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional
packetStartDateTime	The packetStartDateTime must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd'T'HH:mm:ss.SSSZ**). ▪ If not supplied, will default to 00:00:00 of yesterday's date. ▪ The maximum number of days between the packetStartDateTime and packetEndDateTime cannot exceed 90 days. ▪ The packetStartDateTime must be before or the same as the packetEndDateTime. ▪ The packetStartDateTime cannot be more than 90 days prior to the current date. ▪ The packetStartDateTime cannot be a future date.	Date-Time. Optional
packetEndDateTime	The packetEndDateTime must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd'T'HH:mm:ss.SSSZ**). ▪ If not supplied, will default to 23:59:59 of "Yesterday's" date. ▪ The maxiumum number of days between the packetStartDateTime and packetEndDateTime cannot exceed 90 days.	Date-Time. Optional.

Responses: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 400 – "packetEndDateTime is before packetStartDateTime."
- Error Code 400 – "Date provided is not in a valid format."

- Error Code 400 – “Account name not provided.”
- Error Code 400 – “packetStartDateTime/packetEndDateTime is a future date.”
- Error Code 400 – “Offset value is a negative value.”
- Error Code 400 – “packetStartDateTime is older than 90 days.”

JSON Example: Raw Query Sample Request

```
{
   "rawQuerySample":
   {
      "accountName":"teamrest",
      "packetStartDateTime":" 2017-08-09T10:25:00Z",
      "packetEndDateTime":" 2017-08-11T10:10:00Z",
      "zoneName": "apexcnamedemo1.com."
   }
}
```

Retrieving Raw Query Sample Report

Method and URI:

```
GET https://ai.ultradns.com/requests/{requestID}
```

Parameters: Must contain a *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Raw Query Sample Report Output DTO* in the response body. Each value is comma-separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – “Unauthorized. Token not found, expired or invalid.”
- Error Code 404 – “No report with the given ID was requested before.”

Table 155 Raw Query Sample Report Output DTO

Response Body	Description	Type
accountName	The name of the Account being queried.	String.
udpTcpIndicator	Will return either **UDP** or **TCP**.	String.
tldSldIndicator	Will return Top Level Domain or Server Level Domain value of either **TLD** or **SLD** server.	String.
queryResponseIndicator	The Query response indicator. Will return either **Query** or **Response.**	String.

Response Body	Description	Type
responseMicroSet	Will return either the response in micro seconds or a **NULL** value if a response is not matched to a query.	Integer.
ucapPacketErr	Packet parsing error in ucap. Will return: **0 = No Error** **1 = IP Header error** **2 = DNS Header error** **3 = Record Count error** **4 = Query error** **5 = Answer error** **6 = Auth error** **7 = Additional error**	Integer.
resolverPort	The Resolver port.	Integer.
resolverIP	The Resolver IP in binary. **4** bytes for IPv4, **16** bytes for IPv6.	Integer. IP address.
truncatedIndicator	The Truncation indicator: **1 = Message truncated** **0 = Not truncated**	Integer.
recursionDesired	The Recursion desired indicator. **1=True** **2=False**	Integer.
recursionAvailable	The Recursive available indicator. **1=True** **2=False**	Integer.
rcode	The Response Code: **0 = NOERROR** **1 = FORMERR** **2 = SERVFAIL** **3 = NXDOMAIN** **4 = NOTIMP** **5 = REFUSED**	Integer.
qdcount	The query count.	Integer.
packetId	Packet identifier. This unique identifier is produced in the capture application from a timestamp, node, and sequence number.	Integer.
packetDatetime	Packet date/time. A timestamp when the packet was received by the capture process.	Date-Time.

Response Body	Description	Type
opcode	The query code: **0 = QUERY** **1 = IQUERY** **2 = STATUS**	Integer.
dnsMsgLength	Length of the DNS message. Excludes ether/IP/UDP/TCP headers.	Integer.
dnsId	The DNS Identifier. An ID generated by the client and returned by the resolver.	Integer.
clientPort	The client port.	Integer.
clientIp	Client IP in either binary. **4** bytes for IPv4 **16** bytes for IPv6	Integer. IP address
checkingDisabled	Checking disabled indicator: **1 = True** **2 = False**	Integer.
authenticDataIndicator	Authentic data indicator: **1 = True** **2 = False**	Integer.
authenticAnswerIndicator	Authoritative answer indicator: **1 = Authoritative** **0 = Non-Authoritative**	Integer.
arcount	The Additional Record count.	Integer.
ancount	The Answer Record count.	Integer.
nscount	The Authority record count.	Integer.
qname	The query name.	String.
qtype	The Query type. **1 = A** **5 = CNAME** **6 = SOA** **28 = AAAA** etc.	Integer.
zoneName	The Zone Name being queried.	String.
ipVersion	The IP version being used by the zone. **IPv4 = 4** **IPv6 = 6**	Integer.

JSON Example: Retrieving Raw Query Sample Report

```
{
    "accountName": "teamrest",
    "udpTcpIndicator": "UDP",
    "tldSldIndicator": "SLD",
```

```
        "queryResponseIndicator": "Query",
        "responseMicroSec": null,
        "ucapPacketErr": 0,
        "resolverPort": 53,
        "resolverIp": "10.31.147.7",
        "truncatedIndicator": 0,
        "recursionDesired": 1,
        "recursionAvailable": 0,
        "rcode": 0,
        "qdcount": 1,
        "packetId": "6267557462119846686",
        "packetDateTime": 2016-03-29T19:21:22.769Z,
        "opcode": 0,
        "dnsMessageLength": 47,
        "dnsId": "64586",
        "clientPort": 53194,
        "clientIp": "10.33.162.158",
        "checkingDisabled": 0,
        "authenticDataIndicator": 1,
        "authenticAnswerIndicator": 0,
        "arcount": 1,
        "ancount": 0,
        "nscount": 0,
        "qname": "apexcnamedemo1.com.",
        "qtype": "A",
        "zoneName": "apexcnamedemo1.com.",
        "ipVersion": "4"
    }
```


The Raw Query Sample Report can be returned in the .CSV format. See the *Calling the APIs* section for further details.

.CSV Example: Raw Query Sample Report


```
Account Name,UDP / TCP,TLD / SLD,Query / Response,Response Time (in µs),Ucap
Packet Error,Resolver Port,Response IP,Truncated Indicator,Recursion
Desired,Recursion Available,Response code,Question Record count,Packet
Id,Packet Date Time,Opcode,DNS Message Length,DNS Id,Client Port,Client
IP,Checking Disabled,Authentic Data Indicator,Authentic Answer
Indicator,Additional Record Count,Answer Record Count,Namespace Record
Count,QName,QType,Zone Name,IP Version
GTV8,UDP,SLD,Query,,0,53,204.74.108.1,0,1,0,0,1,6524704301611519106,2018-02-
20T18:25:02.298Z,0,32,60455,39787,107.21.211.150,0,0,0,0,0,0,gmon-
n.invalid.,SOA,gmon-n.invalid.,4
GTV8,UDP,SLD,Response,163,0,53,204.74.108.1,0,1,0,0,1,6524704301611519269,201
8-02-20T18:25:02.298Z,0,143,60455,39787,107.21.211.150,0,0,1,0,1,1,gmon-
n.invalid.,SOA,gmon-n.invalid.,4
GTV8,UDP,SLD,Query,,0,53,204.74.108.1,0,1,0,0,1,6524704305906574096,2018-02-
20T18:25:03.385Z,0,32,37375,46677,107.21.211.150,0,0,0,0,0,0,gmon-
n.invalid.,SOA,gmon-n.invalid.,4
```

Advanced Response Codes Report

The Advanced Response codes report shows the DNS return codes for zones. This report can indicate a trend in DNS return codes, or pinpoint where or when specific DNS return codes began occurring in responses.

Requesting Advanced Reponse Codes Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/advanced_response_codes?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Advanced Response Codes Parameters

Table 156 Advanced Response Codes Parameters DTO

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain *Advanced Response Codes DTO.*

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – “Unauthorized. Token not found, expired or invalid.”
- Error Code 400 – “reportEndDate is before reportStartDate.”
- Error Code 400 – “Date provided is not in a valid format.”
- Error Code 400 – “Account name not provided.”
- Error Code 400 – “reportStartDate / reportEndDate is a future date.”
- Error Code 400 – “Offset value is a negative value.”

- Error Code 400 – “reportStartDate is older than 90 days.”

Advanced Response Codes Report DTO

Table 157 Advanced Response Codes DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
zoneName	The results of the one zone being returned. ▪ When not specified, all zones under the account will be queried for reporting. ▪ Wildcards in the zone name are currently not supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday’s date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday’s date.	Date. Optional.

JSON Example: Requesting Advanced Response Codes Report

```
{
    "advancedResponseCodes" : {
            "accountName": "teamrest",
            "reportStartDate": "2017-05-20",
            "reportEndDate": "2017-07-20",
            "zoneName": "apexcnamedemo1.com."
      }
}
```

Retrieving Advanced Response Codes Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: Must contain a *ReportRequest DTO*.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Advanced Response Codes Report Output DTO* in the response body. Each value is comma-separated.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 158 Advanced Response Codes Report Output DTO

Response Body	Description	Type
responseCount	The total sum of all of the returned "count" fields returned as an integer value. (i.e. refusedCount, notimpCount, nxdomainCount etc)	Long.
reportStartDate	The date from which the report aggregation begins for the zone.	Date.
reportEndDate	The date on which the aggregation ends for the zone	Date.
refusedCount	The total count of Queries refused.	Long.
notimpCount	The total count of Not implemented results.	Long.
nxdomainCount	The total count of Non-existent domains returned.	Long.
servfailCount	The total count of Server failures returned..	Long.
formerrCount	The total count of Format errors returned.	Long.
noerrorCount	The total count of No errors returned.	Long.
accountName	The name of the Account being queried.	String.
zoneName	The Zone Name being queried.	String.

JSON Example: Retrieving Advanced Response Codes Report

```
[
    {
        "responseCount": 1,
        "reportStartDate": "2016-03-29",
```

```
        "reportEndDate": "2016-03-29",
        "refusedCount": 0,
        "notimpCount": 0,
        "nxdomainCount": 0,
        "servfailCount": 1,
        "formerrCount": 0,
        "noerrorCount": 0,
        "accountName": "teamrest",
        "zoneName": "apexcnamedemo1.com."
    }
]
```


.CSV Example: Advanced Response Codes Report


```
Total Response Count,Report Start Date,Report End Date,Refused Count,Not
Implemented Count,Nxdomain Count,Service fail Count,Formerr Count,No Error
Count,Account Name,Zone Name
11084,2018-02-20,2018-05-07,0,0,0,0,0,11084,GTV8,gmon-a.invalid.
```

Host Level Advanced Response Codes

The Host Level Advanced Response codes report shows the DNS return codes for hosts. This report can indicate a trend in DNS return codes, or pinpoint where or when specific DNS return codes began occurring in responses.

Requesting Host Level Advanced response Codes Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/advanced_response_codes/host?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Host Level Advanced Response Codes Parameters

Table 159 Host Level Advanced Response Codes Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain *Host Level Advanced Response Codes Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 –"Unauthorized. Token not found, expired or invalid."
- Error Code 400 – reportEndDate is before reportStartDate.
- Error Code 400 – Date provided is not in a valid format.
- Error Code 400 - Zone name is not provided.
- Error Code 400 – reportStartDate / reportEndDate is a future date.

- Error Code 400 – Offset value is a negative value.
- Error Code 400 – reportStartDate is older than 90 days.

Host Level Advanced Response Codes Report DTO

Table 160 Host Level Advanced Response Codes DTO

Field	Description	Type
accountName	The name of the account.	String. Optional.
zoneName	The results of the one zone being returned. ▪ Wildcards in the zone name are currently not supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Required.
hostName	The name of the host to be returned. ▪ If not supplied, defaults to “All” hosts in the zone. ▪ The HostName is a FQDN.	String. Optional
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday’s date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday’s date.	Date. Optional.

JSON Example: Requesting Host Level Advanced Response Codes Report

```
{
   "hostAdvancedResponseCodes" : {
         "accountName":"teamrest",
         "zoneName": "apexcnamedemo1.com.",
         "hostName": "www.apexcnamedemo1.com",
         "reportStartDate": "2016-07-20",
```

```
        "reportEndDate":"2016-08-20"
        }
}
```

Retrieving Host Level Advanced Response Codes Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: Must contain a Report Request ID DTO.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Host Level Advanced Response Codes Report Output DTO* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 161 Host Level Advanced Response Codes Report Output DTO

Response Body	Description	Type
zoneName	The Zone Name being queried.	String.
hostName	The Host Name (under the Zone) being queried.	String.
accountName	The name of the Account being queried.	String.
reportStartDate	The start date that the report is being run from.	Local Date.
reportEndDate	The end date that the report is being run up to.	Local Date.
responseCount	The total count of responses found.	Long.
noerrorCount	The total count of no errors found.	Long.
formerrCount	The total count of format errors found.	Long.
servfailCount	The server failures count found.	Long.
nxdomainCount	The non-existent domains count found.	Long.

Response Body	Description	Type
notimpCount	The not implemented count found.	Long.
refusedCount	The Query refused count found.	Long.

JSON Example: Retrieving the Host Level Advanced Response Codes Report


```
[
    {
        "zoneName": "apexcnamedemo1.com.",
        "hostName": "h1.apexcnamedemo1.com.",
        "accountName": "teamrest",
        "reportStartDate": "2018-04-19",
        "reportEndDate": "2018-04-23",
        "responseCount": 2,
        "noerrorCount": 0,
        "formerrCount": 0,
        "servfailCount": 2,
        "nxdomainCount": 0,
        "notimpCount": 0,
        "refusedCount": 0
    },
    {
        "zoneName": "apexcnamedemo1.com.",
        "hostName": "h2.apexcnamedemo1.com.",
        "accountName": "teamrest",
        "reportStartDate": "2018-04-27",
        "reportEndDate": "2018-04-27",
        "responseCount": 1,
        "noerrorCount": 0,
        "formerrCount": 0,
        "servfailCount": 1,
        "nxdomainCount": 0,
        "notimpCount": 0,
        "refusedCount": 0
    },
    {
        "zoneName": "apexcnamedemo1.com.",
        "hostName": "h3.apexcnamedemo1.com.",
        "accountName": "teamrest",
        "reportStartDate": "2018-04-26",
        "reportEndDate": "2018-04-26",
        "responseCount": 1,
        "noerrorCount": 0,
        "formerrCount": 0,
        "servfailCount": 1,
        "nxdomainCount": 0,
        "notimpCount": 0,
        "refusedCount": 0
    }
]
```

.CSV Example: Host Level Advanced Response Codes

```
Zone Name,Host Name,Account Name,Report Start Date,Report End Date,Total
Response Count,No Error Count,Formerr Count,Service fail Count,Nxdomain
Count,Not Implemented Count,Refused Count
apexcnamedemo1.com.,h1.apexcnamedemo1.com.,teamrest,2018-04-19,2018-04-
23,2,0,0,2,0,0,0
apexcnamedemo1.com.,h2.apexcnamedemo1.com.,teamrest,2018-04-27,2018-04-
27,1,0,0,1,0,0,0
apexcnamedemo1.com.,h3.apexcnamedemo1.com.,teamrest,2018-04-26,2018-04-
26,1,0,0,1,0,0,0
```

Volume Change Report

The Volume Changes report indicates the change in volume by Zone from the previous month's volume, and can also provide from the previous 3 months and 12 months as well. This report can help identify trends in volumes for a zone and pinpoint in which month volumes increased or decreased. You can then use other reports for determine a cause.

Requesting Volume Change Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/volume_change?offset={offset}
&limit={limit}
```

Parameters: Can include the following:

Volume Change Report Parameters

Table 162 Volume Change Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain *Volume Change Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 –"Unauthorized. Token not found, expired or invalid."
- Error Code 400 – "Account name is not valid."
- Error Code 400 – Offset value is a negative value.

Volume Change Report DTO

Table 163 Volume Change Report DTO

Field	Description	Type
accountName	The name of the account.	String. Optional.
zoneName	The specific zone name under the account. ▪ Wildcards in the zone name are currently not supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.

JSON Example: Requesting Volume Change Report

```
{
"volumeChange" : {
      "accountName":"vchangerpt",
      "zoneName": "soaptest.biz"
      }
}
```

Retrieving Volume Change Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: Must contain a *Report Request ID DTO*.

Body: None

Response: If task completes, Status Code 200 OK is returned with a *Volume Change Report Response DTO* in the response body.

Errors: An error code is returned under the following conditions:

- Error Code 401 – “Unauthorized. Token not found, expired or invalid.”
- Error Code 404 – “No report with the given ID was requested before.”

Table 164 Volume Change Report Response DTO

Field	Description	Type
accountName	The account name for which the report is being generated.	String.
zoneName	The Zone Name that is being queried.	String.
accountRank	The Account Rank is designated in part by the volume of change in a zone. A lower the rank number (1), the higher the volume of change in the zone.	Integer.

Field	Description	Type
numberOfPriorMonths	The number of previous months in relation to the current calculated response counts.	Integer.
priorMonths	The full month name and year of the prior month(s) in relation to when the report is being run. If the numberOfPriorMonths is greater than one, this will be returned as "% {number of months}". For example, "%12 months" for the last 12 months.	String.
monthId	The month and year formatted as yyyymm. For example, 201805 (2018 May)	Long.
priorResponseCount	The total response count from the previous month(s).	Long.
responseCount	The total response count of the current month.	Long.

JSON Example: Retrieving the Volume Change Report


```
[
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 0,
        "numberOfPriorMonths": 1,
        "priorMonths": "July 2017",
        "monthId": 201707,
        "priorResponseCount": 5663422,
        "responseCount": 5894257
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "August 2017",
        "monthId": 201708,
        "priorResponseCount": 5894257,
        "responseCount": 5704277
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "September 2017",
        "monthId": 201709,
        "priorResponseCount": 5704277,
        "responseCount": 5616032
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
```

```
        "priorMonths": "October 2017",
        "monthId": 201710,
        "priorResponseCount": 5616032,
        "responseCount": 205062851
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "November 2017",
        "monthId": 201711,
        "priorResponseCount": 205062851,
        "responseCount": 158920372
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "December 2017",
        "monthId": 201712,
        "priorResponseCount": 158920372,
        "responseCount": 166914338
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "January 2018",
        "monthId": 201801,
        "priorResponseCount": 166914338,
        "responseCount": 158644273
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "February 2018",
        "monthId": 201802,
        "priorResponseCount": 158644273,
        "responseCount": 163176019
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "March 2018",
        "monthId": 201803,
        "priorResponseCount": 163176019,
        "responseCount": 136838033
    },
    {
        "accountName": "teamrest",
```

```
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "April 2018",
        "monthId": 201804,
        "priorResponseCount": 136838033,
        "responseCount": 142039030
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "May 2018",
        "monthId": 201805,
        "priorResponseCount": 142039030,
        "responseCount": 166970835
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 1,
        "priorMonths": "June 2018",
        "monthId": 201806,
        "priorResponseCount": 166970835,
        "responseCount": 151441808
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 3,
        "priorMonths": "%3 Month",
        "monthId": 201806,
        "priorResponseCount": 136838033,
        "responseCount": 151441808
    },
    {
        "accountName": "teamrest",
        "zoneName": "vcreport.biz.",
        "accountRank": 1,
        "numberOfPriorMonths": 12,
        "priorMonths": "%12 Month",
        "monthId": 201806,
        "priorResponseCount": 5663422,
        "responseCount": 151441808
    }
]
```


.CSV Example: Retrieving the Volume Change Report


```
Account Name,Zone Name,Account Rank,Number Of Prior Months,Prior Months,Month
Id,Prior Response Count,Response Count
teamrest,vcreport.biz.,1,1,August 2017,201708,5894257,5704277
teamrest,vcreport.biz.,1,1,September 2017,201709,5704277,5616032
teamrest,vcreport.biz.,1,1,October 2017,201710,5616032,205062851
```

```
teamrest,vcreport.biz.,1,1,November 2017,201711,205062851,158920372
teamrest,vcreport.biz.,1,1,December 2017,201712,158920372,166914338
teamrest,vcreport.biz.,1,1,January 2018,201801,166914338,158644273
teamrest,vcreport.biz.,1,1,February 2018,201802,158644273,163176019
teamrest,vcreport.biz.,1,1,March 2018,201803,163176019,136838033
teamrest,vcreport.biz.,1,1,April 2018,201804,136838033,142039030
teamrest,vcreport.biz.,1,1,May 2018,201805,142039030,166970835
```

Class C Network Level Directional Response Counts Report

The Class C Network Level Directional Response Counts Report displays the number of responses sent to Class C networks.

Requesting Class C Network Level Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/class_c_network?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Class C Network Level Directional Response Counts Report Parameters

Table 165 Class C Network Level Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain the *Class C Network Level Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 –"Unauthorized. Token not found, expired or invalid."
- Error Code 400 – reportEndDate is before reportStartDate.
- Error Code 400 – Date provided is not in a valid format.
- Error Code 400 - Account name is not provided.
- Error Code 400 – reportStartDate / reportEndDate is a future date.

- Error Code 400 – Offset value is a negative value.
- Error Code 400 – reportStartDate is older than X days.

Class C Network Level Directional Response Counts Report DTO

Table 166 Class C Network Level Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
zoneName	The zone for which the report is being produced for. ▪ Wildcards in the zone name are currently not supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date.	Date. Optional.
classCNetwork	The Class C Network from which the dns queries originated.	String. Optional.
country	The country from which the dns queries originated.	String. Optional.
city	The city from which the dns queries originated.	String. Optional.
region	The region from which the dns queries originated.	String. Optional.

JSON Example: Requesting Class C Network Level Directional Response Counts Report

```
{
   "classCNetworkDirectionalResponseCounts": {
         "accountName": "soaptest",
         "zoneName": apexcnamedemo1.com"
         "reportStartDate":"2018-03-28",
         "reportEndDate":"2018-05-28",
         "classCNetwork": "50.203.91.0",
         "country": "united states",
         "city": "ashburn",
         "region": "virgina"
   }
}
```

Retrieving Class C Network Level Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Class C Network Level Directional Counts Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – “Unauthorized. Token not found, expired or invalid.”
- Error Code 404 – “No report with the given ID was requested before.”

Table 167 Class C Network Level Directional Counts Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
classCNetwork	The Class C Network from which the dns queries originated.	String.
country	The Country from which the dns queries originated.	String.
city	The City (from the above country) from which the dns queries originated.	String.
region	The Region from which the dns queries originated.	String.

Response Body	Description	Type
authNode	The Authoritative DNS Resolution Node from which the answers originated. These are displayed as airport codes that correspond to the node.	String.
responseCount	The total response count grouped by classCNetwork, country, city, region, and authNode.	Long

Response Link Headers

Table 168 Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/class_c_network?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/class_c_network?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Class C Network Level Directional Response Counts Report

```
[
    {
        "accountName": "soaptest",
        "reportStartDate": "2018-04-23",
        "reportEndDate": "2018-04-23",
        "classCNetwork": "10.31.147.0",
        "country": "united states",
        "city": "ashburn",
        "region": "virginia",
        "authNode": "IAD",
        "responseCount": 2000
    },
  {
        "accountName": "soaptest",
        "reportStartDate": "2018-04-23",
```

```
        "reportEndDate": "2018-04-23",
        "classCNetwork": "10.31.147.0",
        "country": "united states",
        "city": "dallas",
        "region": "texas",
        "authNode": "DFW",
        "responseCount": 4000
    }
]
```


.CSV Example: Retrieving the Class C Network Level Directional Response Counts Report


```
Account Name,Report Start Date,Report End Date,Class C Network,Country,City,
Region,Authoritative DNS Node,Total Response Count
soaptest,2018-04-23,2018-04-23,10.31.147.0,united
states,ashburn,virginia,IAD,0
```

Client IP Directional Response Counts Report

The Client IP Directional Response Counts Report displays the number of responses sent to Client IPs.

Requesting Client IP Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/client_ip?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Client IP Directional Response Counts Report Parameters

Table 169 Client IP Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain the *Client IP Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – When account name is not provided.
- Error Code 400 – When Client Class C is not provided.
- Error Code 400 – If date provided is not in valid format.
- Error Code 400 – If reportEndDate is before reportStartDate.

- Error Code 400 – If reportStartDate or reportEndDate is a future date.
- Error Code 400 – If reportStartDate is older than 90 days.
- Error Code 400 – If offset is a negative value.

Client IP Directional Response Counts Report DTO

Table 170 Client IP Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
classCNetwork	The Class C Network from which the dns queries originated.	String. Required.
zoneName	The results for the one zone that is being returned. ▪ Wildcards in the zone name are not currently supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportEndDate cannot be a future date.	Date. Optional.
clientIP	The Client IP from which the dns queries originated.	String. Optional.
country	The country from which the dns queries originated.	String. Optional.

Field	Description	Type
region	The region from which the dns queries originated.	String. Optional.
city	The city from which the dns queries originated.	String. Optional.

JSON Example: Requesting Client IP Directional Response Counts Report

```
{
"clientIPDirectionalResponseCounts": {
   "accountName": "NameOfTheAccount",
   "classCNetwork": "10.33.162.0",
   "reportStartDate": "2018-06-05",
   "reportEndDate": "2018-09-01"
   }
}
```

Retrieving Client IP Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Client IP Directional Response Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 171 Client IP Directional Response Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
classCNetwork	The Class C Network from which the dns queries originated.	String.
clientIP	The Client IP from which the dns queries originated.	String.

Response Body	Description	Type
country	The Country from which the dns queries originated.	String.
city	The City (from the above country) from which the dns queries originated.	String.
region	The Region from which the dns queries originated.	String.
authNode	The Authoritative DNS Resolution Node from which the answers originated. These are displayed as airport codes that correspond to the node.	String.
responseCount	The total response count grouped by classCNetwork, country, city, region, and authNode.	Long

Response Link Headers

Table 172 Response Links Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/client_ip?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/client_ip?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Client IP Directional Response Counts Report

```
{
      "accountName": "teamrest",
      "reportStartDate": "2018-08-22",
      "reportEndDate": "2018-09-01",
      "classCNetwork": "10.33.162.0",
      "clientIP": "10.33.162.158",
```

```
      "country": "united states",
      "city": "ashburn",
      "region": "virginia",
      "authNode": "IAD",
      "responseCount": 300
},
{
      "accountName": "teamrest",
      "reportStartDate": "2018-08-22",
      "reportEndDate": "2018-08-22",
      "classCNetwork": "10.33.162.0",
      "clientIP": "10.33.162.159",
      "country": "united states",
      "city": "frederick",
      "region": "maryland",
      "authNode": "IAD",
      "responseCount": 100
},
{
      "accountName": "teamrest",
      "reportStartDate": "2018-08-30",
      "reportEndDate": "2018-08-30",
      "classCNetwork": "10.33.162.0",
      "clientIP": "10.33.162.160",
      "country": "united states",
      "city": "baltimore",
      "region": "maryland",
      "authNode": "IAD",
      "responseCount": 100
}
```


.CSV Example: Retrieving the Client IP Directional Response Counts Report


```
Account Name,Report Start Date,Report End Date,Class C Network,Client
IP,Country,City,Region,Authoritative DNS Node,Total Response Count
teamrest,2018-08-22,2018-09-01,10.33.162.0,10.33.162.158,united
states,ashburn,virginia,IAD,300
teamrest,2018-08-22,2018-08-22,10.33.162.0,10.33.162.159,united
states,frederick,maryland,IAD,100
teamrest,2018-08-30,2018-08-30,10.33.162.0,10.33.162.160,united
states,baltimore,maryland,IAD,100
```

Zone Directional Response Counts Report

The Zone Directional Response Counts Report displays the number of responses sent for zones from a specified region.

Requesting Zone Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/zone?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Zone Directional Response Counts Report Parameters

Table 173 Zone Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
oimit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain the *Zone Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – When account name is not provided.
- Error Code 400 – When Client Class C is not provided.
- Error Code 400 – If date provided is not in valid format.
- Error Code 400 – If reportEndDate is before reportStartDate.

- Error Code 400 – If reportStartDate or reportEndDate is a future date.
- Error Code 400 – If reportStartDate is older than 90 days.
- Error Code 400 – If offset is a negative value.

Zone Directional Response Counts Report DTO

Table 174 Zone Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
classCNetwork	The Class C Network from which the dns queries originated.	String. Required.
zoneName	The results for the one zone that is being returned. ▪ Wildcards in the zone name are not currently supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportEndDate cannot be a future date.	Date. Optional.
Country	The country from which the dns queries originated.	String. Optional.
Region	The region from which the dns queries originated.	String. Optional.
City	The city from which the dns queries originated.	String. Optional.

JSON Example: Requesting Zone Directional Response Counts Report

```
{
"zoneDirectionalResponseCounts": {
   "accountName": "NameOfTheAccount",
   "classCNetwork": "10.33.162.0",
   "reportStartDate": "2018-06-05",
   "reportEndDate": "2018-09-01"
   }
}
```

Retrieving Zone Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Zone Directional Response Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 175 Zone Directional Response Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
classCNetwork	The Class C Network from which the dns queries originated.	String.
zoneName	The Zone that was queried from the given region (country + city + region combination) and the given authNode.	String.
country	The Country from which the dns queries originated.	String.
city	The City (from the above country) from which the dns queries originated.	String.
region	The Region from which the dns queries originated.	String.

Response Body	Description	Type
authNode	The Authoritative DNS Resolution Node from which the answers originated. These are displayed as airport codes that correspond to the node.	String.
responseCount	The total response count grouped by classCNetwork, country, city, region, and authNode.	Long

Response Link Headers

Table 176 Response Links Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/zone ?offset=8limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/zone ?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Zone Directional Response Counts Report

```
{
      "accountName": "teamrest",
      "reportStartDate": "2018-08-22",
      "reportEndDate": "2018-09-01",
      "classCNetwork": "10.33.162.0",
      "zoneName": "zoneName.com.",
      "country": "united states",
      "city": "ashburn",
      "region": "virginia",
      "authNode": "IAD",
      "responseCount": 300
},
{
      "accountName": "teamrest",
```

```
      "reportStartDate": "2018-08-22",
      "reportEndDate": "2018-08-22",
      "classCNetwork": "10.33.162.0",
      "zoneName": "zoneName.com.",
      "country": "united states",
      "city": "frederick",
      "region": "maryland",
      "authNode": "IAD",
      "responseCount": 100
},
{
      "accountName": "teamrest",
      "reportStartDate": "2018-08-30",
      "reportEndDate": "2018-08-30",
      "classCNetwork": "10.33.162.0",
      "zoneName": "zoneName.com.",
      "country": "united states",
      "city": "baltimore",
      "region": "maryland",
      "authNode": "IAD",
      "responseCount": 100
}
```


.CSV Example: Retrieving the Zone Directional Response Counts Report


```
Account Name,Report Start Date,Report End Date,Class C Network,Zone
Name,Country,City,Region,Authoritative DNS Node,Total Response Count
teamrest,2018-08-22,2018-09-01,10.33.162.0,zoneName.com.,united
states,ashburn,virginia,IAD,300
teamrest,2018-08-22,2018-08-22,10.33.162.0,zoneName.com.,united
states,frederick,maryland,IAD,100
teamrest,2018-08-30,2018-08-30,10.33.162.0,zoneName.com.,united
states,baltimore,maryland,IAD,100
```

Host Directional Response Counts Report

The Host Directional Response Counts Report displays the number of responses sent for hosts from a specified region.

The Host Directional Response Counts Report now contains two new parameters: **zoneNames** and **wrap**. When these two features are provided, up to ten million records can be returned, and will provide source ip details at the host level. If wrap is not provided, the Host Directional Response Counts Report will return the results as normal.

Requesting Host Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/host?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Host Directional Response Counts Report Parameters

Table 177 Host Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain the *Host Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – When account name is not provided.

- Error Code 400 – When Client Class C is not provided.
- Error Code 400 – If date provided is not in valid format.
- Error Code 400 – If reportEndDate is before reportStartDate.
- Error Code 400 – If reportStartDate or reportEndDate is a future date.
- Error Code 400 – If reportStartDate is older than 90 days.
- Error Code 400 – If offset is a negative value.

Host Directional Response Counts Report DTO

Table 178 Host Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
classCNetwork	The Class C Network from which the dns queries originated.	String. Optional.
zoneName	The results for the one zone that is being returned. ▪ Wildcards in the zone name are not currently supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Required.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportEndDate cannot be a future date.	Date. Optional.
hostName	The results for the host that is being returned, if specified. (geo ip information)	String. Optional.

Field	Description	Type
	▪ Wildcards in the zone name are not currently supported.	
country	The country from which the dns queries originated.	String. Optional.
region	The region from which the dns queries originated.	String. Optional.
city	The city from which the dns queries originated.	String. Optional.
zoneNames	The list of zones for which the report is being generated. The maximum number of zones allowed in this field is 1,000. Either the zoneName or zoneNames parameter need to be included in the DTO body.	List <string>. Required.
wrap	The wrap parameter is used to return a large number of records in a single response. The maximum number of records that can be returned at once is ten million. If set to **True**, up to ten million records will be returned. If the response counts is larger than ten million, the Task-Status will return an Error. Additionally, when set to True, the report will only be available in .CSV format. If set to **False**, the existing behavior of offset and limit will apply.	Boolean. Optional.

JSON Example: Requesting Host Directional Response Counts Report

```
{
"hostDirectionalResponseCounts": {
   "accountName": "NameOfTheAccount",
   "classCNetwork": "10.33.162.0",
   "zoneName": "apexcnamedemo.com."
   "reportStartDate": "2018-06-05",
   "reportEndDate": "2018-09-01"
   }
}
```

Retrieving Host Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Host Directional Response Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 179 Host Directional Response Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
classCNetwork	The Class C Network from which the dns queries originated.	String.
zoneName	The Zone that was queried from the given region (country + city + region combination).	String.
hostname	The host that was queried from the given region (country + city + region combination) and the given authNode.	String.
country	The Country from which the dns queries originated.	String.
city	The City (from the above country) from which the dns queries originated.	String.
region	The Region from which the dns queries originated.	String.
authNode	The Authoritative DNS Resolution Node from which the answers originated. These are displayed as airport codes that correspond to the node.	String.
responseCount	The total response count grouped by classCNetwork, country, city, region, and authNode.	Long.
hostType	The type of the host. Will only be returned if wrap is set to True.	Integer.
sourceIP	The source IP from which the dns queries originated. Will only be returned if wrap is set to True.	String.

Response Link Headers

Table 180 Response Links Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/host?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/host?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Host Directional Response Counts Report

```
[
   {
        "accountName": "teamrest",
        "reportStartDate": "2018-10-03",
        "reportEndDate": "2018-10-03",
        "classCNetwork": "10.33.162.0",
        "zoneName": "apexcnamedemo.com.",
        "hostName": "a1.apexcnamedemo.com.",
        "country": "united states",
        "city": "ashburn",
        "region": "virginia",
        "authNode": "IAD",
        "responseCount": 100
   }
]
```

.CSV Example: Retrieving the Host Directional Response Counts Report

```
Account Name,Report Start Date,Report End Date,Class C Network,Zone Name,Host
Name,Country,City,Region,Authoritative DNS Node,Total Response Count
teamrest,2018-10-03,2018-10-03,10.33.162.0, apexcnamedemo.com.,
a2.apexcnamedemo.com.,united states,ashburn,virginia,IAD,100
```

.CSV Example: Retrieving the Host Directional Response Counts Report using WRAP

```
Account Name,Report Start Date,Report End Date,Class C Network,Zone Name,Host
Name,Country,City,Region,Authoritative DNS Node,Total Response Count, Host
Type, Source IP
AccountName,2021-02-22,2021-02-
22,127.0.0.0,apexcnamedemo.com,,argentina,buenos aires,ciudad de buenos
aires,,0,0,127.0.0.1
AccountName,2021-02-18,2021-02-
23,10.33.162.0,apexcnamedemo.com,,argentina,buenos aires,ciudad de buenos
aires,,0,0,10.33.162.159
```

Postal Code Directional Response Counts Report

The Postal Code Directional Response Counts Report displays the postal code from which the DNS query originates.

Requesting Postal Code Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/postal_code?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Postal Code Directional Response Counts Report Parameters

Table 181 Postal Code Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 10,000 records.	Integer. Optional.

Body: Must contain the *Postal Code Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – When account name is not provided.
- Error Code 400 – When Client Class C is not provided.
- Error Code 400 – If date provided is not in valid format.
- Error Code 400 – If reportEndDate is before reportStartDate.

- Error Code 400 – If reportStartDate or reportEndDate is a future date.
- Error Code 400 – If reportStartDate is older than 90 days.
- Error Code 400 – If offset is a negative value.

Postal Code Directional Response Counts Report DTO

Table 182 Postal Code Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account.	String. Required.
classCNetwork	The Class C Network from which the dns queries originated.	String. Required.
country	The country from which the dns queries originated.	String. Required.
zoneName	The results for the one zone that is being returned. ▪ Wildcards in the zone name are not currently supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ If not provided, will default to yesterday's date. ▪ The reportEndDate cannot be a future date.	Date. Optional.
postalCode	▪ The postal code from which the DNS query originated.	String. Optional.
region	The region from which the dns queries originated.	String. Optional.

Field	Description	Type
city	The city from which the dns queries originated.	String. Optional.

JSON Example: Requesting Postal Code Directional Response Counts Report

```
{
"postalCodeDirectionalResponseCounts": {
   "accountName": "NameOfTheAccount",
   "classCNetwork": "10.33.162.0",
   "country": "united states",
   "reportStartDate": "2018-06-05",
   "reportEndDate": "2018-09-01"
   }
}
```

Retrieving Postal Code Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Postal Code Directional Response Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 183 Postal Code Directional Response Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
classCNetwork	The Class C Network from which the dns queries originated.	String.
zoneName	The Zone that was queried from the given region (country + city + region combination).	String.
country	The Country from which the dns queries originated.	String.

Response Body	Description	Type
city	The City (from the above country) from which the dns queries originated.	String.
region	The Region from which the dns queries originated.	String.
postalCode	The postal code from which the DNS query originated.	String.
authNode	The Authoritative DNS Resolution Node from which the answers originated. These are displayed as airport codes that correspond to the node.	String.
responseCount	The total response count grouped by classCNetwork, country, city, region, and authNode.	Long

Response Link Headers

Table 184 Response Links Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/postal_code?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/postal_code?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Postal Code Directional Response Counts Report

```
[
   {
        "accountName": "teamrest",
        "reportStartDate": "2018-10-03",
        "reportEndDate": "2018-10-03",
        "classCNetwork": "10.33.162.0",
        "zoneName": "apexcnamedemo.com.",
        "country": "united states",
```

```
        "city": "ashburn",
        "region": "virginia",
        "postalCode": "20166",
        "authNode": "IAD",
        "responseCount": 100
    }
]
```


.CSV Example: Retrieving the Postal Code Directional Response Counts Report


```
Account Name,Report Start Date,Report End Date,Class C Network,Zone
Name,Country,City,Region,Postal Code,Authoritative DNS Node,Total Response
Count
teamrest,2018-10-03,2018-10-03,10.33.162.0,apexcnamedemo.com.,united
states,ashburn,virginia,20166,IAD,100
```

Country Code Directional Response Counts Report

The Country Code Directional Response Counts Report displays the country codes from which the DNS queries originate.

Requesting Country Code Directional Response Counts Report

Method and URI:

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/country_code?offset={offset}&limit={limit}
```

Parameters: Can include the following:

Country Code Directional Response Counts Report Parameters

Table 185 Country Code Directional Response Counts Report Parameters

Parameter	Description	Type
offset	This field is optional. If not specified, initial records will always be returned specified to the limit. This parameter allows pagination on the reporting records retrieved. The offset will be the integer value that specifies the position of first result to be retrieved. Specify offset as 0 for the first results to be retrieved.	Integer. Optional.
limit	This field is optional. If not specified, the total number of records returned in the response will be equal to the default value 1000. This parameter allows pagination on the reporting records retrieved. The maximum number of results to be retrieved in a single response is 100,000 records.	Integer. Optional.

Body: Must contain the *Country Code Directional Response Counts Report DTO*.

Response: If task completes, Status Code 201 is returned with a requestId in the response body.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – If date provided is not in valid format.
- Error Code 400 – If reportEndDate is before reportStartDate.
- Error Code 400 – If reportStartDate or reportEndDate is a future date.
- Error Code 400 – If reportStartDate is older than 90 days.

- Error Code 400 – If offset is a negative value.

Country Code Directional Response Counts Report DTO

Table 186 Country Code Directional Response Counts Report DTO

Field	Description	Type
accountName	The name of the account. If not provided, the report will consist of details for all of the zones within the account(s) that the user has access to.	String. Optional.
zoneName	The results for the one zone that is being returned. ▪ Wildcards in the zone name are not currently supported. ▪ Zone names with and without a DOT(.) at the end are supported.	String. Optional.
reportStartDate	The reportStartDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ If not provided, will default to yesterday's date. ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ The reportStartDate must be before or the same as the reportEndDate. ▪ The reportStartDate cannot be more than 90 days prior to the current date. ▪ The reportStartDate cannot be a future date.	Date. Optional
reportEndDate	The reportEndDate must be supplied in the ISO 8601 UTC format (**yyyy-MM-dd**). ▪ If not provided, will default to yesterday's date. ▪ The maximum number of days between the reportStartDate and reportEndDate cannot exceed 90 days. ▪ The reportEndDate cannot be a future date.	Date. Optional.

JSON Example: Requesting Country Code Directional Response Counts Report

```
POST
https://api.ultradns.com/reports/dns_resolution/directional_response_counts/country_code?offset={0}&limit={100000}
{
"countryCodeDirectionalResponseCounts" : {
"accountName" : "NameOfTheAccount",
"reportStartDate":"2019-07-28",
"reportEndDate":"2019-07-28"
   }
```

```
}
```

Retrieving Country Code Directional Response Counts Report

Method and URI:

```
GET https://api.ultradns.com/requests/{requestID}
```

Parameters: *Report Request ID DTO*

Body: None

Response: If task completes, Status Code 200 OK is returned with a list of *Country Code Directional Response Report Response DTO*.

Errors: An error code is returned under the following conditions:

- Error Code 401 – "Unauthorized. Token not found, expired or invalid."
- Error Code 404 – "No report with the given ID was requested before."

Table 187 Country Code Directional Response Report Response DTO

Response Body	Description	Type
accountName	The Account Name for which the report is being generated for.	String.
reportStateDate	The start date that the report is being run from.	Date.
reportEndDate	The end date that the report is being run to.	Date.
zoneName	The zone name for which the report is being generated, or the zone name that is present under the specified account.	String.
responseCountTotal	The total response count for queries that originated from all countries for the requested report inputs.	Long
responseCountOther	The total response count for queries that originated from countries other than ones specifically returned in the response, as well as the Caribbean Netherlands.	Long.
responseCountByCountryCode	The total response count by country code, as a Map. Refer to *Table 188* for a full list of Country Codes.	Map (String, Long)

Table 188 Country Code Directional Response Counts - Country Codes

Country Code as String	Response Count as Long
AD	Response count for queries that originated from Andorra.
AE	Response count for queries that originated from the United Arab Emirates.

Country Code as String	Response Count as Long
AF	Response count for queries that originated from Afghanistan.
AG	Response count for queries that originated from Antigua and Barbuda.
AI	Response count for queries that originated from Anguilla.
AL	Response count for queries that originated from Albania.
AM	Response count for queries that originated from Armenia.
AN	Response count for queries that originated from Netherland Antilles.
AO	Response count for queries that originated from Angola.
AQ	Response count for queries that originated from Antarctica.
AR	Response count for queries that originated from Argentina.
AS	Response count for queries that originated from American Samoa.
AT	Response count for queries that originated from Austria.
AU	Response count for queries that originated from Australia.
AW	Response count for queries that originated from Aruba.
AX	Response count for queries that originated from the Aland Islands.
AZ	Response count for queries that originated from Azerbaijan.
BA	Response count for queries that originated from Bosnia-Herzegovina.
BB	Response count for queries that originated from Barbados.
BD	Response count for queries that originated from Bangladesh.
BE	Response count for queries that originated from Belgium.
BF	Response count for queries that originated from Burkina Faso.
BG	Response count for queries that originated from Bulgaria.
BH	Response count for queries that originated from Bahrain.
BI	Response count for queries that originated from Burundi.
BJ	Response count for queries that originated from Benin.
BL	Response count for queries that originated from Saint Barthelemy.
BM	Response count for queries that originated from Bermuda.
BN	Response count for queries that originated from Brunei Darussalam.
BO	Response count for queries that originated from Bolivia.
BQ	Response count for queries that originated from the Dutch Caribbean.
BR	Response count for queries that originated from Brazil.
BS	Response count for queries that originated from the Bahamas.
BT	Response count for queries that originated from Bhutan.
BV	Response count for queries that originated from Bouvet Island.
BW	Response count for queries that originated from Botswana.
BY	Response count for queries that originated from Belarus.
BZ	Response count for queries that originated from Belize.
CA	Response count for queries that originated from Canada.
CC	Response count for queries that originated from the Cocos (Keeling) Islands.

Country Code as String	Response Count as Long
CD	Response count for queries that originated from the Democratic Republic of the Congo.
CF	Response count for queries that originated from the Central African Republic.
CG	Response count for queries that originated from Congo.
CH	Response count for queries that originated from Switzerland.
CI	Response count for queries that originated from Cote d'Ivoire.
CK	Response count for queries that originated from the Cook Islands.
CL	Response count for queries that originated from Chile.
CM	Response count for queries that originated from Cameroon.
CN	Response count for queries that originated from China.
CO	Response count for queries that originated from Colombia.
CR	Response count for queries that originated from Costa Rica.
CU	Response count for queries that originated from Cuba.
CV	Response count for queries that originated from Cape Verde.
CW	Response count for queries that originated from Curacao.
CX	Response count for queries that originated from Christmas Island.
CY	Response count for queries that originated from Cyprus.
CZ	Response count for queries that originated from the Czech Republic.
DE	Response count for queries that originated from Germany.
DJ	Response count for queries that originated from Djibouti.
DK	Response count for queries that originated from Denmark.
DM	Response count for queries that originated from Dominica.
DO	Response count for queries that originated from the Dominican Republic.
DZ	Response count for queries that originated from Algeria.
EC	Response count for queries that originated from Ecuador.
EE	Response count for queries that originated from Estonia.
EG	Response count for queries that originated from Egypt.
EH	Response count for queries that originated from Western Sahara.
ER	Response count for queries that originated from Eritrea.
ES	Response count for queries that originated from Spain.
ET	Response count for queries that originated from Ethiopia.
FI	Response count for queries that originated from Finland.
FJ	Response count for queries that originated from Fiji.
FK	Response count for queries that originated from the Falkland Islands.
FM	Response count for queries that originated from the Federated States of Micronesia.
FO	Response count for queries that originated from the Faroe Islands.
FR	Response count for queries that originated from France.

Country Code as String	Response Count as Long
GA	Response count for queries that originated from Gabon.
GB	Response count for queries that originated from the United Kingdom - England, Northern Ireland, Scotland, and Wales.
GD	Response count for queries that originated from Grenada.
GE	Response count for queries that originated from Georgia.
GF	Response count for queries that originated from French Guiana.
GG	Response count for queries that originated from Guernsey.
GH	Response count for queries that originated from Ghana.
GI	Response count for queries that originated from Gibraltar.
GL	Response count for queries that originated from Greenland.
GM	Response count for queries that originated from Gambia.
GN	Response count for queries that originated from Guinea.
GP	Response count for queries that originated from Guadeloupe.
GQ	Response count for queries that originated from Equatorial Guinea.
GR	Response count for queries that originated from Greece.
GS	Response count for queries that originated from South Georgia and the South Sandwich Islands.
GT	Response count for queries that originated from Guatemala.
GU	Response count for queries that originated from Guam.
GW	Response count for queries that originated from Guinea-Bissau.
GY	Response count for queries that originated from Guyana.
HK	Response count for queries that originated from Hong Kong.
HM	Response count for queries that originated from Heard Island and the McDonald Islands.
HN	Response count for queries that originated from Honduras.
HR	Response count for queries that originated from Croatia.
HT	Response count for queries that originated from Haiti.
HU	Response count for queries that originated from Hungary.
ID	Response count for queries that originated from Indonesia.
IE	Response count for queries that originated from Ireland.
IL	Response count for queries that originated from Israel.
IM	Response count for queries that originated from the Isle of Man.
IN	Response count for queries that originated from India.
IO	Response count for queries that originated from the British Indian Ocean Territory - Chagos Islands.
IQ	Response count for queries that originated from Iraq.
IR	Response count for queries that originated from Iran.
IS	Response count for queries that originated from Iceland.
IT	Response count for queries that originated from Italy.

Country Code as String	Response Count as Long
JE	Response count for queries that originated from Jersey.
JM	Response count for queries that originated from Jamaica.
JO	Response count for queries that originated from Jordan.
JP	Response count for queries that originated from Japan.
KE	Response count for queries that originated from Kenya.
KG	Response count for queries that originated from Kyrgyzstan.
KH	Response count for queries that originated from Cambodia.
KI	Response count for queries that originated from Kiribati.
KM	Response count for queries that originated from Comoros.
KN	Response count for queries that originated from St. Kitts and Nevis.
KP	Response count for queries that originated from the Democratic Peoples's Republic of Korea.
KR	Response count for queries that originated from the Republic of Korea.
KW	Response count for queries that originated from Kuwait.
KY	Response count for queries that originated from the Cayman Islands.
KZ	Response count for queries that originated from Kazakhstan.
LA	Response count for queries that originated from Lao People's Democratic Republic.
LB	Response count for queries that originated from Lebanon.
LC	Response count for queries that originated from St. Lucia.
LI	Response count for queries that originated from Liechtenstein.
LK	Response count for queries that originated from Sri Lanka.
LR	Response count for queries that originated from Liberia.
LS	Response count for queries that originated from Lesotho.
LT	Response count for queries that originated from Lithuania.
LU	Response count for queries that originated from Luxembourg.
LV	Response count for queries that originated from Latvia.
LY	Response count for queries that originated from the Libyan Arab Jamahiriya.
MA	Response count for queries that originated from Morocco.
MC	Response count for queries that originated from Monaco.
MD	Response count for queries that originated from the Republic of Moldova.
ME	Response count for queries that originated from Montenegro.
MF	Response count for queries that originated from Saint Martin.
MG	Response count for queries that originated from Madagascar.
MH	Response count for queries that originated from the Marshall Islands.
MK	Response count for queries that originated from Macedonia, the former Republic of Yugoslav.
ML	Response count for queries that originated from Mali.
MM	Response count for queries that originated from Myanmar.

Country Code as String	Response Count as Long
MN	Response count for queries that originated from Mongolia.
MO	Response count for queries that originated from Macao.
MP	Response count for queries that originated from the Commonwealth of the Northern Mariana Islands.
MQ	Response count for queries that originated from Martinique.
MR	Response count for queries that originated from Mauritania.
MS	Response count for queries that originated from Montserrat.
MT	Response count for queries that originated from Malta.
MU	Response count for queries that originated from Mauritius.
MV	Response count for queries that originated from Maldives.
MW	Response count for queries that originated from Malawi.
MX	Response count for queries that originated from Mexico.
MY	Response count for queries that originated from Malaysia.
MZ	Response count for queries that originated from Mozambique.
NA	Response count for queries that originated from Namibia.
NC	Response count for queries that originated from New Caledonia.
NE	Response count for queries that originated from Niger.
NF	Response count for queries that originated from Norfolk Island.
NG	Response count for queries that originated from Nigeria.
NI	Response count for queries that originated from Nicaragua.
NL	Response count for queries that originated from the Netherlands.
NO	Response count for queries that originated from Norway.
NP	Response count for queries that originated from Nepal.
NR	Response count for queries that originated from Nauru.
NU	Response count for queries that originated from Niue.
NZ	Response count for queries that originated from New Zealand.
OM	Response count for queries that originated from Oman.
PA	Response count for queries that originated from Panama.
PE	Response count for queries that originated from Peru.
PF	Response count for queries that originated from French Polynesia.
PG	Response count for queries that originated from Papua New Guinea.
PH	Response count for queries that originated from the Philippines.
PK	Response count for queries that originated from Pakistan.
PL	Response count for queries that originated from Poland.
PM	Response count for queries that originated from Saint Pierre and Miquelon.
PN	Response count for queries that originated from Pitcairn.
PR	Response count for queries that originated from Puerto Rico.
PS	Response count for queries that originated from (Occupied) Palestinian Territory.

Country Code as String	Response Count as Long
PT	Response count for queries that originated from Portugal.
PW	Response count for queries that originated from Palau.
PY	Response count for queries that originated from Paraguay.
QA	Response count for queries that originated from Qatar.
RE	Response count for queries that originated from Reunion.
RO	Response count for queries that originated from Romania.
RS	Response count for queries that originated from Serbia.
RU	Response count for queries that originated from the Russian Federation.
RW	Response count for queries that originated from Rwanda.
SA	Response count for queries that originated from Saudi Arabia.
SB	Response count for queries that originated from the Solomon Islands.
SC	Response count for queries that originated from Seychelles.
SD	Response count for queries that originated from Sudan.
SE	Response count for queries that originated from Sweden.
SG	Response count for queries that originated from Singapore.
SH	Response count for queries that originated from St Helena.
SI	Response count for queries that originated from Slovenia.
SJ	Response count for queries that originated from Svalbard and Jan Mayen.
SK	Response count for queries that originated from Slovakia.
SL	Response count for queries that originated from Sierra Leone.
SM	Response count for queries that originated from San Marino.
SN	Response count for queries that originated from Senegal.
SO	Response count for queries that originated from Somalia.
SR	Response count for queries that originated from Suriname.
SS	Response count for queries that originated from South Sudan.
ST	Response count for queries that originated from Sao Tome and Principe.
SV	Response count for queries that originated from El Salvador.
SX	Response count for queries that originated from Saint Maarten.
SY	Response count for queries that originated from the the Syrian Arab Republic.
SZ	Response count for queries that originated from Swaziland.
TC	Response count for queries that originated from the Turks and Caicos Islands.
TD	Response count for queries that originated from Chad.
TF	Response count for queries that originated from the French Southern Territories.
TG	Response count for queries that originated from Togo.
TH	Response count for queries that originated from Thailand.
TJ	Response count for queries that originated from Tajikistan.
TK	Response count for queries that originated from Tokelau.

Country Code as String	Response Count as Long
TL	Response count for queries that originated from the Democratic Republic of Timor-Leste.
TM	Response count for queries that originated from Turkmenistan.
TN	Response count for queries that originated from Tunisia.
TO	Response count for queries that originated from Tonga.
TR	Response count for queries that originated from the Republic of Turkey.
TT	Response count for queries that originated from Trinidad and Tobago.
TV	Response count for queries that originated from Tuvalu.
TW	Response count for queries that originated from Taiwan.
TZ	Response count for queries that originated from the United Republic of Tanzania.
UA	Response count for queries that originated from Ukraine.
UG	Response count for queries that originated from Uganda.
UM	Response count for queries that originated from the United States Minor Outlying Islands
US	Response count for queries that originated from the United States.
UY	Response count for queries that originated from Uruguay.
UZ	Response count for queries that originated from Uzbekistan.
VA	Response count for queries that originated from Vatican City.
VC	Response count for queries that originated from Saint Vincent and the Grenadines.
VE	Response count for queries that originated from the Bolivarian Republic of Venezuela.
VG	Response count for queries that originated from the British Virgin Islands.
VI	Response count for queries that originated from the U.S Virgin Islands.
VN	Response count for queries that originated from Vietnam.
VU	Response count for queries that originated from Vanuatu.
WF	Response count for queries that originated from Wallis and Futuna.
WS	Response count for queries that originated from Samoa.
YE	Response count for queries that originated from Yemen.
YT	Response count for queries that originated from Mayotte.
ZA	Response count for queries that originated from South Africa.
ZM	Response count for queries that originated from Zambia.
ZW	Response count for queries that originated from Zimbabwe.

- Bouvet Island, Dutch Carribean, Netherlands Antilles, and Vatican City are not currently supported at this time, and as such, will always be returned as a 0.
- The following country codes are not recognized by the Country Code Directional Response Counts Report: **AP**, **EU**, and **FX**.

Response Link Headers

Table 189 Response Links Headers

Field	Description
Link	**Relative URL to next page of report if available:** </v1/reports/dns_resolution/directional_response_counts/country_code?offset=8&limit=4>; rel="next" **Relative URL to previous page of report if available:** </v1/reports/dns_resolution/directional_response_counts/country_code?offset=0&limit=4>; rel="previous" When using the next or previous link header to retrieve report data, you must perform another POST call, and include the original body content (if any) and new query parameters (such as offset and limit). When continuing to use subsequent Link Headers to retrieve additional results, you must continue to perform the POST call per link header to retrieve the next set of report details.
Limit	Specify the maximum number of records in requested response. Cannot be greater than maximum allowed limit. Currently maximum allowed limit is 100k.
Results	Total rows in the report response.

JSON Example: Retrieving the Country Code Directional Response Counts Report


```
[
   {
      "accountName": "teamrest",
      "reportStartDate": "2019-07-11",
      "reportEndDate": "2019-07-11",
      "zoneName": "clienitp.com.",
      "responseCountTotal": 200,
      "responseCountOther": 0,
      "responseCountByCountryCode": {
         "AD": 0,
         "AE": 0,
         "AF": 0,
         "AG": 0,
         "AI": 0,
         "AL": 0,
         "AM": 0,
         "AN": 0,
         "AO": 0,
         "AQ": 0,
         "AR": 0,
         "AS": 0,
         "AT": 0,
         "AU": 0,
         "AW": 0,
         "AX": 0,
         "AZ": 0,
         "BA": 0,
```

```
        "BB": 0,
        "BD": 0,
        "BE": 0,
        "BF": 0,
        "BG": 0,
        "BH": 0,
        "BI": 0,
        "BJ": 0,
        "BL": 0,
        "BM": 0,
        "BN": 0,
        "BO": 0,
        "BQ": 0,
        "BR": 0,
        "BS": 0,
        "BT": 0,
        "BV": 0,
        "BW": 0,
        "BY": 0,
        "BZ": 0,
        "CA": 0,
        "CC": 0,
        "CD": 0,
        "CF": 0,
        "CG": 0,
        "CH": 0,
        "CI": 0,
        "CK": 0,
        "CL": 0,
        "CM": 0,
        "CN": 0,
        "CO": 0,
        "CR": 0,
        "CU": 0,
        "CV": 0,
        "CW": 0,
        "CX": 0,
        "CY": 0,
        "CZ": 0,
        "DE": 0,
        "DJ": 0,
        "DK": 0,
        "DM": 0,
        "DO": 0,
        "DZ": 0,
        "EC": 0,
        "EE": 0,
        "EG": 0,
        "EH": 0,
        "ER": 0,
        "ES": 0,
        "ET": 0,
        "FI": 0,
        "FJ": 0,
        "FK": 0,
        "FM": 0,
        "FO": 0,
```

```
        "FR": 0,
        "GA": 0,
        "GB": 0,
        "GD": 0,
        "GE": 0,
        "GF": 0,
        "GG": 0,
        "GH": 0,
        "GI": 0,
        "GL": 0,
        "GM": 0,
        "GN": 0,
        "GP": 0,
        "GQ": 0,
        "GR": 0,
        "GS": 0,
        "GT": 0,
        "GU": 0,
        "GW": 0,
        "GY": 0,
        "HK": 0,
        "HM": 0,
        "HN": 0,
        "HR": 0,
        "HT": 0,
        "HU": 0,
        "ID": 0,
        "IE": 0,
        "IL": 0,
        "IM": 0,
        "IN": 0,
        "IO": 0,
        "IQ": 0,
        "IR": 0,
        "IS": 0,
        "IT": 0,
        "JE": 0,
        "JM": 0,
        "JO": 0,
        "JP": 0,
        "KE": 0,
        "KG": 0,
        "KH": 0,
        "KI": 0,
        "KM": 0,
        "KN": 0,
        "KP": 0,
        "KR": 0,
        "KW": 0,
        "KY": 0,
        "KZ": 0,
        "LA": 0,
        "LB": 0,
        "LC": 0,
        "LI": 0,
        "LK": 0,
        "LR": 0,
```

```
        "LS": 0,
        "LT": 0,
        "LU": 0,
        "LV": 0,
        "LY": 0,
        "MA": 0,
        "MC": 0,
        "MD": 0,
        "ME": 0,
        "MF": 0,
        "MG": 0,
        "MH": 0,
        "MK": 0,
        "ML": 0,
        "MM": 0,
        "MN": 0,
        "MO": 0,
        "MP": 0,
        "MQ": 0,
        "MR": 0,
        "MS": 0,
        "MT": 0,
        "MU": 0,
        "MV": 0,
        "MW": 0,
        "MX": 0,
        "MY": 0,
        "MZ": 0,
        "NA": 0,
        "NC": 0,
        "NE": 0,
        "NF": 0,
        "NG": 0,
        "NI": 0,
        "NL": 0,
        "NO": 0,
        "NP": 0,
        "NR": 0,
        "NU": 0,
        "NZ": 0,
        "OM": 0,
        "PA": 0,
        "PE": 0,
        "PF": 0,
        "PG": 0,
        "PH": 0,
        "PK": 0,
        "PL": 0,
        "PM": 0,
        "PN": 0,
        "PR": 0,
        "PS": 0,
        "PT": 0,
        "PW": 0,
        "PY": 0,
        "QA": 0,
        "RE": 0,
```

```
        "RO": 0,
        "RS": 0,
        "RU": 0,
        "RW": 0,
        "SA": 0,
        "SB": 0,
        "SC": 0,
        "SD": 0,
        "SE": 0,
        "SG": 0,
        "SH": 0,
        "SI": 0,
        "SJ": 0,
        "SK": 0,
        "SL": 0,
        "SM": 0,
        "SN": 0,
        "SO": 0,
        "SR": 0,
        "SS": 0,
        "ST": 0,
        "SV": 0,
        "SX": 0,
        "SY": 0,
        "SZ": 0,
        "TC": 0,
        "TD": 0,
        "TF": 0,
        "TG": 0,
        "TH": 0,
        "TJ": 0,
        "TK": 0,
        "TL": 0,
        "TM": 0,
        "TN": 0,
        "TO": 0,
        "TR": 0,
        "TT": 0,
        "TV": 0,
        "TW": 0,
        "TZ": 0,
        "UA": 0,
        "UG": 0,
        "UM": 0,
        "US": 200,
        "UY": 0,
        "UZ": 0,
        "VA": 0,
        "VC": 0,
        "VE": 0,
        "VG": 0,
        "VI": 0,
        "VN": 0,
        "VU": 0,
        "WF": 0,
        "WS": 0,
        "YE": 0,
```

```
            "YT": 0,
            "ZA": 0,
            "ZM": 0,
            "ZW": 0
        }
    }
]
```

.CSV Example: Retrieving the Country Code Directional Response Counts Report

```
Account Name,Report Start Date,Report End Date,Zone
Name,Total,Other,AD,AE,AF,AG,AI,AL,AM,AN,AO,AQ,AR,AS,AT,AU,AW,AX,AZ,BA,BB,BD,
BE,BF,BG,BH,BI,BJ,BL,BM,BN,BO,BQ,BR,BS,BT,BV,BW,BY,BZ,CA,CC,CD,CF,CG,CH,CI,CK
,CL,CM,CN,CO,CR,CU,CV,CW,CX,CY,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,EH,ER,ES,ET,FI,F
J,FK,FM,FO,FR,GA,GB,GD,GE,GF,GG,GH,GI,GL,GM,GN,GP,GQ,GR,GS,GT,GU,GW,GY,HK,HM,
HN,HR,HT,HU,ID,IE,IL,IM,IN,IO,IQ,IR,IS,IT,JE,JM,JO,JP,KE,KG,KH,KI,KM,KN,KP,KR
,KW,KY,KZ,LA,LB,LC,LI,LK,LR,LS,LT,LU,LV,LY,MA,MC,MD,ME,MF,MG,MH,MK,ML,MM,MN,M
O,MP,MQ,MR,MS,MT,MU,MV,MW,MX,MY,MZ,NA,NC,NE,NF,NG,NI,NL,NO,NP,NR,NU,NZ,OM,PA,
PE,PF,PG,PH,PK,PL,PM,PN,PR,PS,PT,PW,PY,QA,RE,RO,RS,RU,RW,SA,SB,SC,SD,SE,SG,SH
,SI,SJ,SK,SL,SM,SN,SO,SR,SS,ST,SV,SX,SY,SZ,TC,TD,TF,TG,TH,TJ,TK,TL,TM,TN,TO,T
R,TT,TV,TW,TZ,UA,UG,UM,US,UY,UZ,VA,VC,VE,VG,VI,VN,VU,WF,WS,YE,YT,ZA,ZM,ZW
teamrest,2019-07-11,2019-07-
11,clienitp.com.,200,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,200,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Usage Summary Report

The Usage Summary Report displays peak data statistics for an account for the last thirty-six months. Each month that is returned consists of domains counts, record type counts, and query statisistics for the given account.

Requesting the Usage Summary Report

Method and URI:

```
GET https://api.ultradns.com/reports/dns/usage_summary
```

Parameters: Must include the following:

Usage Summary Report Parameters

Table 190 Usage Summary Report Parameters

Parameter	Description	Type
accountName	The Account name for which the Usage Summary Report is being requested.	String. Required.

Body: None.

Response: If task completes, Status Code 200 is returned with the *Usage Summary Report Output DTO*.

Table 191 Usage Summary Report Output DTO

Parameter	Description	Type
UsageSummary	The list of usage summaries for the previous three years (thirty-six months), beginning from the current month.	Array.

Errors: An error is returned under the following conditions:

- Error Code 401 – Unauthorized. Token not found, expired or invalid.
- Error Code 400 – accountName is not provided.
- Error Code 400 – You do not have access to the accountName.

Usage Summary Report DTO

Table 192 Usage Summary DTO

Field	Description	Type
accountName	The Account associated to the Usage Summary row.	String.
month	The month and year for which the returned statistics apply.	String.
domainsCount	The peak (highest) number of Domains that existed under the account name for the givenmonth.	Long.
recordsCount	The peak (highest) number of Records that existed under the account name for the given month.	Long.
queryResponsesCount	The total number of DNS query responses that were served for the account name during the given month.	Long.
urlForwardCount	The peak (highest) number of URL Forwards that existed under the account name for the given month.	Long.
sitebackerRecordsCount	The peak (highest) number of SiteBacker Records that existed under the account name for the given month.	Long.
trafficControllerRecordsCount	The peak (highest) number of Traffic Controller Records that existed under the account name for the given month.	Long.
directionalRecordsCount	The peak (highest) number of Directional Pool Records that existed under the account name for the given month.	Long.

JSON Example: Usage Summary Report Response


```
{
    "usageSummary": [
        {
            "accountName": "GTV8",
            "month": "January 2019",
            "domainsCount": 1863,
            "recordsCount": 10530,
            "queryResponsesCount": 1176520,
            "urlForwardCount": 35,
            "siteBackerRecordsCount": 112,
            "trafficControllerRecordsCount": 56,
            "directionalRecordsCount": 115
        },
        {
            "accountName": "GTV8",
            "month": "December 2018",
            "domainsCount": 1863,
            "recordsCount": 10524,
            "queryResponsesCount": 5464627,
```

```
            "urlForwardCount": 35,
            "siteBackerRecordsCount": 112,
            "trafficControllerRecordsCount": 56,
            "directionalRecordsCount": 120
        },
        {
            "accountName": "GTV8",
            "month": "November 2018",
            "domainsCount": 1862,
            "recordsCount": 10497,
            "queryResponsesCount": 293095094,
            "urlForwardCount": 35,
            "siteBackerRecordsCount": 111,
            "trafficControllerRecordsCount": 56,
            "directionalRecordsCount": 121
        },
        {
            "accountName": "GTV8",
            "month": "October 2018",
            "domainsCount": 1861,
            "recordsCount": 10478,
            "queryResponsesCount": 200952818,
            "urlForwardCount": 35,
            "siteBackerRecordsCount": 108,
            "trafficControllerRecordsCount": 56,
            "directionalRecordsCount": 121
        },
        {
            "accountName": "GTV8",
            "month": "September 2018",
            "domainsCount": 1858,
            "recordsCount": 10433,
            "queryResponsesCount": 176045732,
            "urlForwardCount": 35,
            "siteBackerRecordsCount": 107,
            "trafficControllerRecordsCount": 56,
            "directionalRecordsCount": 120
        },
```


.CSV Example: Retrieving the Usage Summary Report


```
Account Name,Month,Domains Count,Records Count,Query Responses
Count,URLForward Count,EmailForward Count,SiteBacker Records
Count,TrafficController Records Count,Directional Records Count
GTV8,April 2019,1941,11130,268949749,51,3,117,56,124
GTV8,March 2019,1938,11017,231447640,51,3,117,56,121
GTV8,February 2019,1934,10925,70017129,50,3,119,57,119
GTV8,January 2019,1885,10624,5758084,43,3,114,57,115
GTV8,December 2018,1863,10524,5464627,35,3,112,56,120
GTV8,November 2018,1862,10497,293095094,35,3,111,56,121
GTV8,October 2018,1861,10478,200952818,35,3,108,56,121
GTV8,September 2018,1858,10433,176045732,35,3,107,56,120
GTV8,August 2018,1852,10377,223102752,34,3,107,55,120
GTV8,July 2018,1851,10364,198165947,34,3,107,55,120
GTV8,June 2018,1849,10344,172711920,34,3,104,54,119
GTV8,May 2018,1847,10339,154026378,33,3,99,55,111
GTV8,April 2018,1845,10321,151449195,31,3,90,55,111
```

```
GTV8,March 2018,1842,10323,166979368,31,3,93,55,110
GTV8,February 2018,1833,10279,142047507,31,3,93,53,110
GTV8,January 2018,1827,10197,136846008,30,3,85,53,110
GTV8,December 2017,1837,10213,163184279,30,3,82,52,110
GTV8,November 2017,1884,10364,158652931,30,3,82,53,105
GTV8,October 2017,1883,10365,166924285,30,3,85,52,104
GTV8,September 2017,1882,10349,158928386,30,3,85,50,104
GTV8,August 2017,1881,10338,205067634,30,3,80,47,104
GTV8,July 2017,1880,10302,5620738,30,3,78,43,101
GTV8,June 2017,1879,10278,5716107,30,3,75,44,101
GTV8,May 2017,1878,10256,5901808,30,3,72,44,101
GTV8,April 2017,1877,10235,5670492,29,3,72,44,100
GTV8,March 2017,1875,10152,5840992,27,3,73,43,101
GTV8,February 2017,1868,10072,4553925,25,3,65,36,98
GTV8,January 2017,1862,10024,4689695,20,3,65,36,97
GTV8,December 2016,1857,9963,4721596,20,3,66,36,94
GTV8,November 2016,1852,9925,5357160,20,3,71,36,97
GTV8,October 2016,1846,9890,5771759,20,3,69,36,87
GTV8,September 2016,1842,9860,7761472,20,3,68,36,72
GTV8,August 2016,1833,9781,8015008,19,3,68,36,69
GTV8,July 2016,1815,9586,8022261,19,3,70,35,44
GTV8,June 2016,1792,9478,7606860,18,3,69,32,42
GTV8,May 2016,1780,9431,7613712,18,3,70,28,41
```

Probe Result Summary Report

Unlike the various reports already documented, the **Probe Result Summary Report** does not require the usage of the Report RequestID to generate the request for the report, and subsequently the return of the report. When a Probe Result Summary Report is requested, the detailed report is returned to the user immediately.

Requesting Probe Result Summary Report

Method and URI:

```
GET https://api.ultradns.com/reports/traffic_services/probe_result/summary
```

Parameters: Must contain a *Probe Results Summary Query Parameters*.

Body: None

Probe Results Summary Query Parameters

Table 193 Probe Results Summary Query Parameters

Field	Description	Type
accountName	**Optional.** The Account for which the Probe Result Summary Report is being requested. If not specified, the default output will be all accounts the user has access to.	String.
zoneName	**Required.** The Zone under the account for which the Probe Result Summary Report is being requested.	String.
poolName	**Optional.** The Simple Load Balancing Pool under the Zone, and under the Account specified for which the Probe Result Summary Report is being requested.	String.
trafficServicePoolType	**Required.** The type of Traffic Service Pool for which the Probe Result Summary Report is being requested. If not specified, value will be SIMPLE_LOAD_BALANCING.	SIMPLE_LOAD_BALANCING or SIMPLE_FAILOVER.
reportStartDateTime	**Required.** The StartDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Summary Report is being requested. StartDateTime must not be more than six (6) months old.	Date-time.

Field	Description	Type
	The date range (begin – end date) for the report cannot be greater than seven (7) days.	
reportEndDateTime	**Required.** The EndDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Summary Report is being requested. The date range (begin – end date) cannot be greater than seven (7) days.	Date-time.
offset	**Optional.** The Offset to start from for paginated responses.	Integer.
limit	**Optional.** The number of rows per page for paginated responses. The default is 25 if not specified. The maximum value that can be provided for the limit is 2,000.	Integer.

Response: If task completes, Status code 200 OK is returned with a *Probe Result Summary DTO* and the following data:

Response Body	Description	Type
probeResultSummary	The list of Probe Result Summaries.	Array.
probeResultSummaryCount	The number of Rows in the report.	Long.

Errors: An error code is returned under the following conditions:

- If the reportEndDateTime is earlier than the reportStartDateTime.
- If the duration between reportStartDateTime and reportEndDateTime exceeds 7 days.
- If the reportStartDateTime is older than 6 months.
- If the {accountName} cannot be accessed by the current userName.

Probe Result Summary Output DTO

Table 194 Probe Result Summary DTO

Field	Description	Type
accountName	The Account associated to the Probe Result Summary row.	String.
zoneName	The Zone associated to the Probe Result Summary row.	String.

Field	Description	Type
poolName	The Pool associated to the Probe Result Summary row.	String.
successes	The number of Probes that were successful for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.
failures	The number of Probes that failed for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.
total	The number of Probes that were executed for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.
reportStartDateTime	The StartDateTime from which the report was requested.	Date-Time.
reportEndDateTime	The EndDateTime up to which the report was requested.	Date-Time.
trafficServicePoolType	The type of Traffic Service Pool for which the Probe Result Summary Report was requested.	String.

JSON Example: Probe Result Summary Report Response

```
{
  "probeResultSummary": [
    {
      "accountName": "javauie2e",
      "zoneName": "slb.test.com.",
      "poolName": "test.slb.test.com.",
      "successes": null,
      "failures": 23,
      "total": 23,
      "reportStartDateTime": "2016-06-15T12:01:00.000Z",
      "reportEndDateTime": "2016-06-15T12:59:00.000Z",
      "trafficServicePoolType": "SIMPLE_LOAD_BALANCING"
    }
  ],
  "probeResultSummaryCount": 6
}
```

If a Report's Results exceed 25 records per page, you can use the "Next / Previous" header command to search the additional results. This function is only available for the Probe Result Summary and Probe Result Details reports.

- Copy the information after the "Link→" from the Header section of your response, and then paste it into the call section. The very last word of the call will state whether or not it will return the next 25 results, or the previous 25 results. (If you are on the first page of 25 results, you will not have the previous option.)

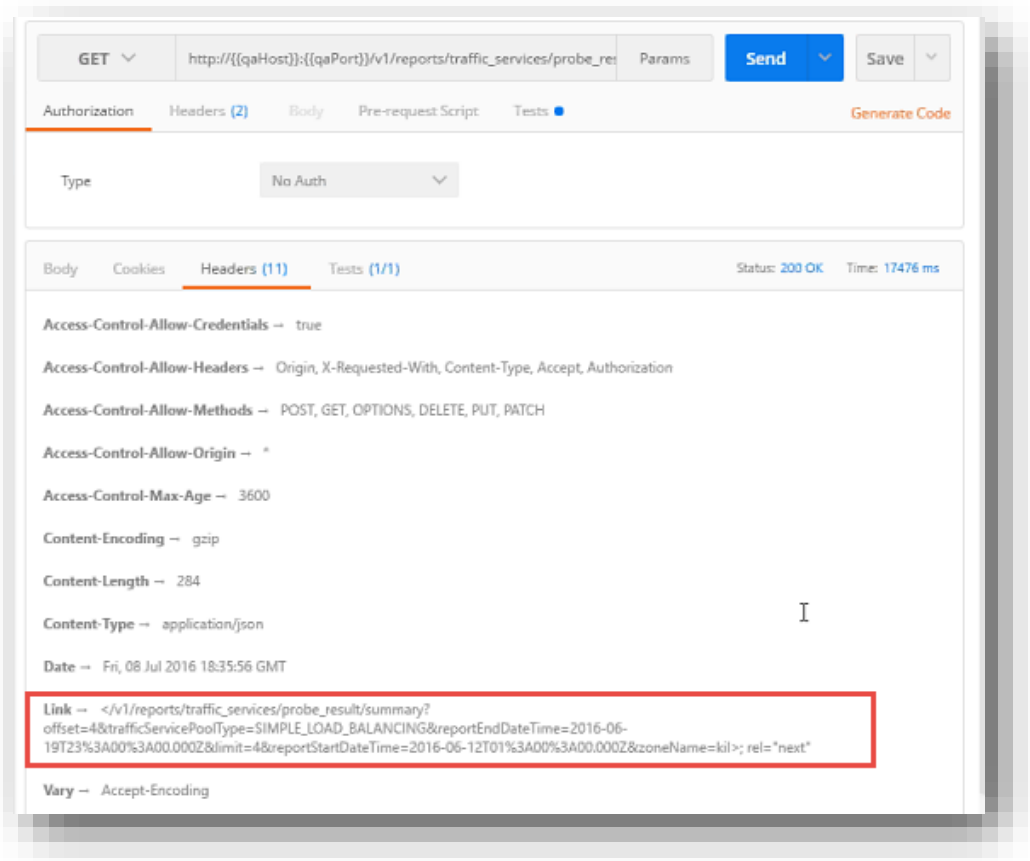

Figure 15 Reporter Service Result - Next/Previous

Probe Result Details Report

Unlike the various reports already documented, the **Probe Result Details Report** does not require the usage of the Report RequestID to generate the request for the report, and subsequently the return of the report. When a Probe Result Details Report is requested, the report is returned to the user immediately.

Requesting Probe Result Details Report

Method and URI:

```
GET https://api.ultradns.com/reports/traffic_services/probe_result/details
```

Body: None

Parameters: Must include *Probe Result Details Query Parameters*.

Probe Result Details Query Parameters

Table 195 Probe Result Details Query Parameters

Field	Description	Type
accountName	**Optional.** The Account for which the Probe Result Details Report is being requested	String.
zoneName	**Required.** The Zone under the account for which the Probe Result Details Report is being requested.	String.
poolName	**Optional.** The Simple Load Balancing Pool under the Zone, and under the Account specified for which the Probe Result Details Report is being requested.	String.
trafficServicePoolType	**Required.** The type of Traffic Service Pool for which the Probe Result Details Report is being requested. If not specified, will be SIMPLE_LOAD_BALANCING.	SIMPLE_LOAD_BALANCING or SIMPLE_FAILOVER.
probeResultType	**Optional.** The type of probe result to view in the report being requested. If not specified, this will be ALL. In the future, the values SUCCESS and FAILURE will be supported.	ALL.
reportStartDateTime	**Required.** The reportStartDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Details Report is being requested.	Date-time.

Field	Description	Type
	StartDateTime must not be more than six (6) months old. The date range (begin – end date) for the report cannot be greater than seven (7) days.	
reportEndDateTime	**Required.** The reportEndDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Details Report is being requested. The date range (begin – end date) cannot be greater than seven (7) days	Date-time.
offset	The Offset to start from for paginated responses.	Integer.
limit	The number of rows per page for paginated responses. The default is 25 if not specified. The maximum value for the limit is 2,000.	Integer.

Response: If task completes, Status Code 200 OK is returned with the *Probe Result Details Output DTO* and the following data:

Response Body	Description	Type
probeResultDetails	The list of Probe Result Details.	Array.
probeResultDetailsCount	The number of Rows in the report.	Long.

Errors: An error code is returned under the following conditions:

- If the reportEndDateTime is earlier than the reportStartDateTime.
- If the duration between reportStartDateTime and reportEndDateTime exceeds 7 days.
- If the reportStartDateTime is older than 6 months.
- If the {accountName } cannot be accessed by the current userName.

Probe Results Details Output DTO

Table 196 Probe Result Details Output DTO

Field	Description	Type
accountName	The Account associated to the Probe Result Details row.	String.

Field	Description	Type
zoneName	The Zone associated to the Probe Result Details row.	String.
poolName	The Pool associated to the Probe Result Details row.	String
trafficServicePoolType	The TrafficServicePoolType associated to the Probe Result Details row.	SIMPLE_LOAD_ BALANCING.
httpMethod	The GET or POST method used by the Probe URL.	String.
httpUrl	The HTTP Probe url used to probe the record within the pool.	String.
httpTransmittedData	Transmitted data in the http probe call.	String.
httpSearchString	The string expected to be present in the http probe response.	String.
httpResponseStatus	The actual http probe response code that was received.	String.
httpResponseString	The actual response string received as a result of the http probe call.	String.
probeResultType	The probeResultType from the request. Value of ALL if both a Success and Failure Probe Result Details Report is being requested.	▪ SUCCESS ▪ FAILURE ▪ ALL
reportStartDateTime	The reportStartDateTime in ISO 8601 UTC for the period for which the Probe Result Details report is being requested.	Date-Time.
reportEndDateTime	The reportEndtDateTime in ISO 8601 UTC for the period for which the Probe Result Details report is being requested.	Date-Time.
poolRecord	The record within the pool that was probed.	String.
poolProbeRegion	The region from which the pool record was probed.	String.
probeStartDateTime	The Date and Time when probing began for the record from the probe region.	Date-Time.
probeEndDateTime	The Date and Time when probing ended for the record from the probe region.	Date-Time.
probeResultStatus	The result from the probe, whether it was a success or a failure.	SUCCESS or FAILURE.
probeResult	The response for the probe that determined the probe result.	String.

JSON Example: Probe Result Details Report Response

```
{
  "probeResultDetails": [
    {
      "accountName": "javauie2e",
      "zoneName": "e2e-slb.com.",
```

```
      "poolName": "slbpool1.e2e-slb.com.",
      "httpUrl": "http://www.google.com/",
      "httpTransmittedData": "",
      "httpSearchString": "N/A",
      "httpResponseStatus": "200",
      "httpResponseString": "OK",
      "trafficServicePoolType": "SIMPLE_LOAD_BALANCING",
      "probeResultType": "ALL",
      "reportStartDateTime": "2016-06-14T00:00:00.000Z",
      "reportEndDateTime": "2016-06-16T01:00:00.000Z",
      "poolRecord": "74.125.138.99",
      "poolProbeRegion": "US-EAST",
      "probeStartDateTime": "2016-06-15T08:55:20.253Z",
      "probeEndDateTime": "2016-06-15T08:55:20.312Z",
      "probeResultStatus": "SUCCESS",
      "probeResult": "Success",
      "httpMethod": "GET"
    },
    {
      "accountName": "javauie2e",
      "zoneName": "e2e-slb.com.",
      "poolName": "slbpool1.e2e-slb.com.",
      "httpUrl": "http://www.google.com/",
      "httpTransmittedData": "",
      "httpSearchString": "N/A",
      "httpResponseStatus": "0",
      "httpResponseString": "N/A",
      "trafficServicePoolType": "SIMPLE_LOAD_BALANCING",
      "probeResultType": "ALL",
      "reportStartDateTime": "2016-06-14T00:00:00.000Z",
      "reportEndDateTime": "2016-06-16T01:00:00.000Z",
      "poolRecord": "2.2.2.2",
      "poolProbeRegion": "US-EAST",
      "probeStartDateTime": "2016-06-15T08:49:00.253Z",
      "probeEndDateTime": "2016-06-15T08:49:30.254Z",
      "probeResultStatus": "FAILURE",
      "probeResult": "User timeout caused connection failure.",
      "httpMethod": "GET"
    }
  ],
  "probeResultDetailsCount": 5479
}
```


If a Report's Results exceed 25 records per page, you can use the "Next / Previous" header command to search the additional results. This function is only available for the Probe Result Summary and Probe Result Details reports.

- Copy the information after the "Link→" from the *Header* section of your response, and then paste it into the call section. The very last word of the call will state whether or not it will return the next 25 results, or the previous 25 results. (If you are on the first page of 25 results, you will not have the previous option)

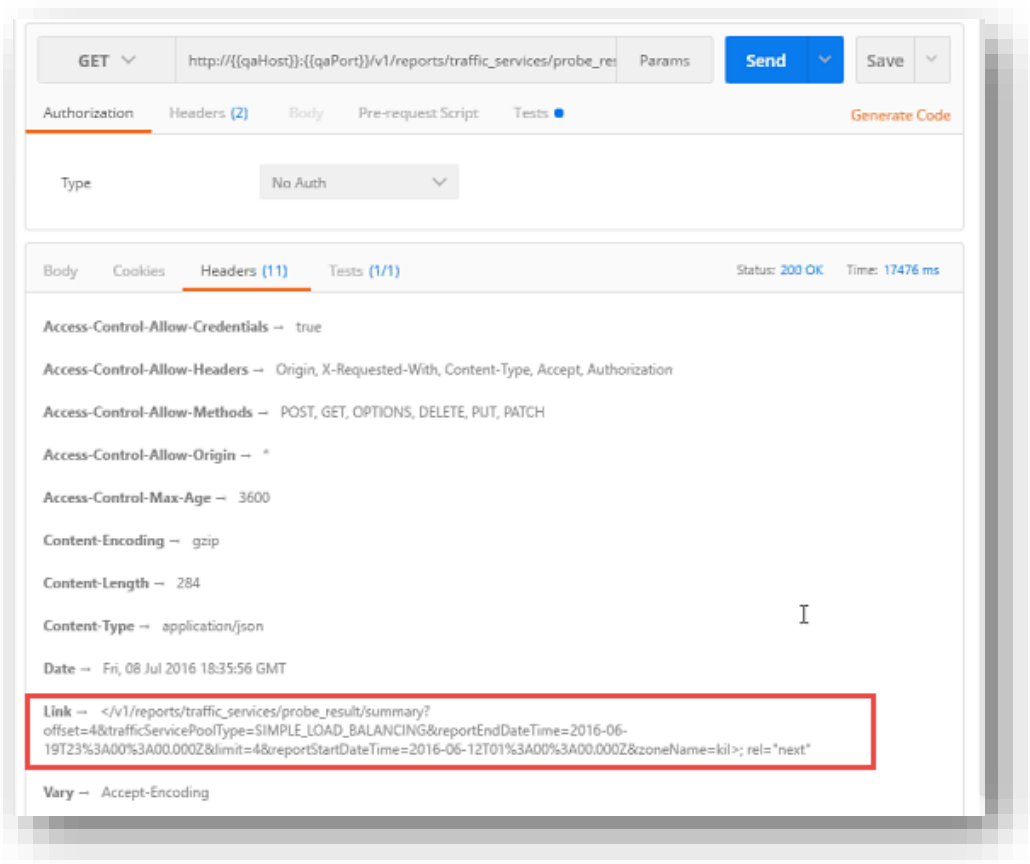

Figure 16 Reporter Service Result - Next/Previous

Audit Log Report

The Audit Log Report returns the results of those entries that are captured in the Auditing process. These events can include the creation, modification, and deletion of a domain or record. The creation or deletion of a user from an account, and even the Change Comment field that can be added to a function for further explanation and description of the event.

Unlike the *Projected Query Volume Report*, *Zone Query Volume Report*, and the *Reporting APIs,* the **Audit Log Report** does not require the usage of the Report RequestID to generate the request for the report, and subsequently the return of the report. When an Audit Log Report is requested, the detailed report is returned to the user immediately.

The Audit Log Report can be returned in a .CSV format, but will require an additional step beyond the default JSON requirements. In the header section, you will need to include the additional field: **Accept**: **text/csv**.

Requesting Audit Log Report

Method and URI:

```
GET https://api.ultradns.com/reports/dns_configuration/audit?filter=any_
search_param&limit=any_integer
```

Body: None

Parameters: Must contain a *Audit Log Query Parameters DTO*.

Audit Log Query Parameters DTO

Table 197 Audit Log Query Parameters

Field	Description	Type
filter	Defines the filter criteria for Audit Log. ▪ **accounts** – Comma-separated list of account names. Default value is **None**, which will return ALL accounts of the logged in user. ▪ **users** - Comma-separated list of user names. Default value is **None**, which will return ALL users. ▪ **change_type** – The type of change to display. Valid values of change_type can be obtained using *Audit Log Query Filters*. Default value is **None**, which will return all change_type values. ▪ **object_type** – The type of object to display. Valid values of object_type can be obtained using *Audit Log Query Filters*. Default value is **None**, which will return all object_type values. ▪ **object_name** – The name of the object. Default value is **None**, which will return all object_name values. ▪ **parent_name** – The name of the parent object. Default value is **None**, which will return all parent_name values. ▪ **date_range** – The specific date range to filter the results by. Date ranges can be provided in the following format: o 15m = 15 minutes from the current date.	String.

Field	Description	Type
	○ 5h = 5 hours from the current date. ○ 30d = 30 days from the current date. ○ 5w = The last 5 weeks from the current date. ○ 1month = The last calendar month from the current date. ○ {start_date – {end_date} = Provide a specific date range in GMT format. (yyyyMMddHHmmss) ▪ **change_comment** – The text fragment of Change Comments search for. Special characters need to be URL encoded. ○ When using a colon (:) as a search paramter for the Audit Log Report, "***/:*** " (slash colon) is required instead of just a colon (:). ○ Search criteria is case sensitive, and will be returned on a partial match (all records containing your search criteria will be returned). ○ Example URL encoded - "Special change @ & # comment" would be Special%20change%20%40%20%26%20%23%20comment	
limit	Allows for pagination of the Audit records received. Maximum value for results received is 250. Minimum is 1. Default value is 50.	Integer.

Response: If task completes, Status Code 200 OKis returned with an *Audit Log Response DTO*, and a response header containing the Link Header having URLS of *next* and *previous* pages as applicable.

Audit Log Response DTO

Table 198 Audit Log Response Parameters

Field	Description	Type
objectType	The type of the object that is being audited.	String.
changeType	Audited action.	String.
object	The name of the object that is being audited.	String.
user	The name of the user who performed the action.	String.
ipAddress	IP address of user or logic who performed the action.	String.
changeTime	Date and Time when action was performed.	String.
account	Account name from which audit was done.	String.
parent	The name of the parent object.	String.
detail	The detail of the audit record.	AuditLogDetail.
changeComment	The details of the most recent comment provided.	String.

Table 199 Audit Log Detail Parameters

Field	Description	Type
changes	The list of details of changes that take place by CRUD operation.	List<AuditLogChangeDetail>.

Table 200 Audit Log Change Detail Parameters

Field	Description	Type
name	The name of the attribute that was added/updated/deleted.	String.
from	The attribute "from" represents the old value of attribute "name" in case of update/delete operation.	String.
to	The attribute "to" represents the new value of attribute "name" in case of update/add operation.	String.

Table 201 Response Headers

Field	Description
Link	**Relative URL to next page of report if available** <v1/reports/dns_configuration/audit?filter=filter_spec&cursor=cursor_spec&Limit=limit_spec&cursorOperation=NEXT>; rel="next" **Relative URL to previous page of report if available** <v1/reports/dns_configuration/audit?filter=filter_spec&cursor=cursor_spec&Limit=limit_spec&cursorOperation=PREVIOUS>; rel="previous"
Limit	The limit specified in audit log query parameter
Results	Total rows in the report page

JSON Example: Audit Log Query Parameters Successful Response

```
{
   "auditRecords":[
      {
         "objectType": "TXT"
         "changeType": "MODIFY"
         "object": "date.gmon-a.invalid."
         "user": "gmonitor"
         "ipAddress": "209.173.57.233"
         "changeTime": "2016-06-12 21:48:02.0"
         "account": "GTV8"
         "parent": null
         "detail": {
             "changes": [
                {
                   "name": "Comments"
                   "from": "Mon Jun 13 03.15.02 2016"
```

```
                    "to": "Mon Jun 13 03.18.02 2016"
                }
            ],
            "others": [0]
        }
    }
  ]
}
```

Errors: An error code is returned under the following conditions:

- If the “limit” value is not between 0 and 250.
- If the “filter” parameter’s value is syntactically incorrect. The value must be a sequence of “key:value” pair where each pair is separated by “::”.
- If invalid filter “key” is supplied in the request.
- If the date range provided in the filter value is not in the format of **“yyyyMMddHHmmss”**.
- If the start date given in the date range is greater than the end date.
- If the logged in user is not authorized for the accounts given in the filter parameter.

JSON Example: Audit Log Example Requests

Example - List latest 50 Audit Records

```
GET https://api.ultradns.com/reports/dns_configuration/audit HTTP/1.1
```

- If filter query parameter is not given then by default last 24h audit records are returned.
- The records retrieved will be sorted in DESC order
- The default values of limit is 50.

Example – List latest 250 Audit Records

```
GET https://api.ultradns.com/reports/dns_configuration/audit?limit=250
```

Example – List Audit Records for a specific date range

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=date_range:20
160302290813-20160402193020
```

Example – List Audit Records for a specific User

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=users:ABLE
```

Example – List Audit Records for a specific object name

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=object_name:1
27.0.0.1
```

Example – List Audit Records for a specific object name and change type

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=object_name:127.0.0.1::change_type:ZONE_TRANSFER_FAILURE
```

Example – List Audit Records for a specific parent name

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=parent_name:test-domain.com.
```

Example – List Audit Records for a specific parent name and change type

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=parent_name:test-domain.com.::change_type:ZONE_TRANSFER_FAILURE
```

Example – List Audit Records for a specific change comment

```
GET
https://api.ultradns.com/reports/dns_configuration/audit?filter=change_Comment:abc
```

Audit Log Query Filters

Method and URI:

```
GET https://api.ultradns.com/reports/dns_configuration/audit/filters
```

Body: None

Parameters: The following Query Parameter will be used:

Table 202 Audit Log Query Parameter

Field	Description	Type
dateRange	The specific date range to filter the results by. Date ranges can be provided in the following format: ▪ 15m = 15 minutes from the current date. ▪ 5h = 5 hours from the current date. ▪ 30d = 30 days from the current date. ▪ 5w = The last 5 weeks from the current date. ▪ 1month = The last calendar month from the current date. {start_date – {end_date} = Provide a specific date range in GMT format. **(yyyyMMddHHmmss)** **Example:** 20160321123010 - 20160401113312	String.

Field	Description	Type
change_Comment	Specify the text fragment to search for. Special characters must be URL encoded. If using the Colon (:) special character, users will need to instead provide " ***/:*** " (slash colon).	String.

Response: If task completes, Status code 200 OK is returned with an *Audit Log Query Filter Response DTO*.

Audit Log Query Filter Response DTO

Table 203 Audit Log Query Filter Parameters

Field	Description	Type
key	Key for audit url. These keys could be value of object_type, change_type etc.	String.
url	The example audit url to search records based the given key.	String.

JSON Example: Audit Log Query Filters Response

```
{
   "filters" : [
      "objectTypes": [
         {
          "key": "A"
          "url": "/reports/dns_configuration/audit?filter=object_type:A"
          },
          {
          "key": "SOA"
          "url": "/reports/dns_configuration/audit?filter=object_type:SOA"
          },
          {
          "key": "NS"
          "url": "/reports/dns_configuration/audit?filter=object_type:NS"
          }
        ],
     "changeTypes": [
          {
          "key": "ADD"
          "url": "/reports/dns_configuration/audit?filter=change_type:ADD"
          },
          {
          "key": "FAILED_LOGIN"
          "url": "/reports/dns_configuration/audit?filter=change_type
                 :FAILED_LOGIN"
          }
        ],
        "users": [
          {
          "key": "gmonitor"
```

```
          "url": "/reports/dns_configuration/audit?filter=user:gmonitor"
          },
          {
           "key": "ketkitest"
           "url": "/reports/dns_configuration/audit?filter=user:ketkitest"
          },
          {
           "key": "sswamy"
           "url": "/reports/dns_configuration/audit?filter=user:sswamy"
          }
        ],
        "accounts": [
          {
           "key": "GTV8"
            "url": "/reports/dns_configuration/audit?filter=account:GTV8"
          },
          {
           "key": "sswamy"
           "url": "/reports/dns_configuration/audit?filter=account:sswamy"
          }
       ]
   ]
}
```


Errors: An error code is returned under the following conditions

- Invalid X-User {userName} supplied in the request header. This user is not linked with any account."

Probe Result Summary v2 Report

Unlike the Probe Result Summary Report, the **Probe Result Summary version 2 Report** has been streamlined to not require as many mandatory fields to generate the request for the report, and subsequently, the return of the report. When a Probe Result Summary v2 Report is requested, the detailed report is returned to the user immediately. *Please note that the **/v2/** path is required in the Method and URI to correctly utilize this report.*

Requesting Probe Result Summary v2 Report

Method and URI:

```
GET https://api.ultradns.com/v2/reports/traffic_services/probe_result/summary
```

Parameters: Must contain a *Probe Results Summary V2 Query Pamameters*.

Body: None

Probe Results Summary v2 Query Parameters

Table 204 Probe Results Summary v2 Query Parameters

Field	Description	Type
accountName	**Required.** The Account for which the Probe Result Summary Report is being requested.	String.
zoneName	**Optional.** The Zone under the account for which the Probe Result Summary Report is being requested.	String.
poolName	**Optional.** The Pool under the Zone, and under the Account specified for which the Probe Result Summary Report is being requested.	String.
trafficServicePoolType	**Optional.** The type of Pool for which the Probe Result Summary Report is being requested. If not specified, value will be ALL.	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER ▪ ALL
probeResultType	**Optional.** The type of probe result to view in the report being requested. If not specified, this will be ALL.	▪ ALL ▪ SUCCESS ▪ FAILURE.
poolRecord	**Optional.** The record within the pool for which the report is being requested.	String

Field	Description	Type
poolRecordType	**Optional.** The type of record within the pool for which the report is being requested. If not specified, this will be ALL.	▪ A ▪ AAAA ▪ CNAME ▪ ALL
poolProbeRegion	**Optional.** The region from which the pool record was probed for report is being requested. If not specified, this will be ALL.	▪ NORTH_AMERICA_EAST ▪ NORTH_AMERICA_WEST ▪ NORTH_AMERICA_CENTRAL ▪ EUROPE_EAST ▪ EUROPE_WEST ▪ SOUTH_AMERICA ▪ ASIA ▪ CHINA ▪ ALL.
reportStartDateTime	**Optional.** The StartDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Summary Report is being requested. StartDateTime must not be more than three (3) months old. The date range (begin – end date) for the report cannot be greater than seven (7) days. If not specified, this will be set to last twenty four (24) hours.	Date-time.
reportEndDateTime	**Optional.** The EndDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Summary Report is being requested. The date range (begin – end date) cannot be greater than seven (7) days.	Date-time.
limit	**Optional.** The number of rows per page for paginated responses. The default is 1,000 if not specified. The maximum value that can be provided for the limit is 1,000.	Integer.

Response: If task completes, Status code 200 OK is returned with a *Probe Result Summary V2 DTO* and the following data:

Response Body	Description	Type
sbtcprobeSummaryRow	The list of Probe Result Summaries.	Array.
sbtcprobeSummaryCount	The number of Rows in the report.	Long.

Errors: An error code is returned under the following conditions:

- If the reportEndDateTime is earlier than the reportStartDateTime.
- If the duration between reportStartDateTime and reportEndDateTime exceeds 7 days.
- If the reportStartDateTime is older than 3 months.
- If the {accountName } cannot be accessed by the current userName.

Probe Result Summary v2 Output DTO

Table 205 Probe Result Summary v2 DTO

Field	Description	Type
accountName	The Account associated to the Probe Result Summary row.	String.
zoneName	The Zone associated to the Probe Result Summary row.	String.
poolName	The Pool associated to the Probe Result Summary row.	String.
poolRecord	The record within the pool that was probed.	String.
poolRecordType	The type of the record within the pool that was probed.	▪ A ▪ AAAA ▪ CNAME
poolProbeRegion	The region from which the pool record was probed.	▪ North America East ▪ North America West ▪ North America Central ▪ Europe East ▪ Europe West ▪ South America ▪ Asia ▪ China
successes	The number of Probes that were successful for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.

Field	Description	Type
failures	The number of Probes that failed for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.
warnings	The number of Probes that were in warning state for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
critical	The number of Probes that were in critical state for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
total	The number of Probes that were executed for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer.
successPercentage	The percentage of Probes that were successful for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
failurePercentage	The percentage of Probes that failed for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
warningPercentage	The percentage of Probes that were in warning state for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
criticalPercentage	The percentage of Probes that were in critical state for the associated Pool Name within the requested report period, from all probe regions for all records under the pool.	Integer
reportStartDateTime	The StartDateTime from which the report was requested.	Date-Time.
reportEndDateTime	The EndDateTime up to which the report was requested.	Date-Time.
trafficServicePoolType	The Pool type associated to the Probe Result Summary row.	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER

JSON Example: Probe Result Summary v2 Report Response


```
{
  "sbtcprobeSummaryRow": [
    {
      "accountName": "javauie2e",
      "zoneName": "slb.test.com.",
      "poolName": "test.slb.test.com.",
      "poolRecord": "www.google.com.",
      "poolRecordType": "CNAME",
      "poolProbeRegion": "North America Central",
      "successes": 60,
      "failures": 0,
      "warnings": 0,
      "critical": 0,
      "total": 60,
      "successPercentage": 100.0,
      "failurePercentage": 0.0,
      "warningPercentage": 0.0,
      "criticalPercentage": 0.0,
      "startDateTime": "2016-06-15T12:01:00.000Z",
      "endDateTime": "2016-06-15T12:59:00.000Z",
      "trafficServicePoolType": "SIMPLE_LOAD_BALANCING"
    }
  ],
  "sbtcprobeSummaryCount": 6
}
```


Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** **GET** <v2/reports/traffic_services/probe_result/summary?accountName=GTV8&cursorOperation=NEXT&limit=50>; rel="next" **Relative URL to previous page of report if available:** <v2/reports/traffic_services/probe_result/summary?accountName=GTV8&cursorOperation=PREVIOUS&limit=50>; rel="previous"

If a Report's Results exceed 1000 records per page, you can use the "Next / Previous" header command to search the additional results.

Probe Result Details v2 Report

Unlike the Probe Result Details Report which is already documented, the **Probe Result Details v2 Report** has been streamlined to not require as many mandatory fields to generate the request for the report, and subsequently, the return of the report When a Probe Result Details v2 Report is requested, the detailed report is returned to the user immediately. *Please note that the **/v2/** path is required in the Method and URI to correctly utilize this report.*

Requesting Probe Result Details v2 Report

Method and URI:

```
GET https://api.ultradns.com/v2/reports/traffic_services/probe_result/details
```

Body: None

Parameters: Must include *Probe Result Details V2 Query Parameters.*

Probe Result Details v2 Query Parameters

Table 206 Probe Result Details v2 Query Parameters

Field	Description	Type
accountName	**Required.** The Account for which the Probe Result Details Report is being requested	String.
zoneName	**Optional.** The Zone under the account for which the Probe Result Details Report is being requested.	String.
poolName	**Optional.** The Pool under the Zone, and under the Account specified for which the Probe Result Summary Report is being requested.	String.
trafficServicePoolType	**Optional.** The type of Pool for which the Probe Result Summary Report is being requested. If not specified, value will be ALL	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER ▪ ALL
poolRecord	**Optional.** The record within the pool for which the report is being requested.	String
poolRecordType	**Optional.** The type of record within the pool for which the report is being requested. If not specified, this will be ALL.	▪ A ▪ AAAA ▪ CNAME ▪ ALL

Field	Description	Type
poolProbeRegion	**Optional.** The region from which the pool record was probed for report is being requested. If not specified, this will be ALL.	▪ NORTH_AMERICA_EAST ▪ NORTH_AMERICA_WEST ▪ NORTH_AMERICA_CENTRAL ▪ EUROPE_EAST ▪ EUROPE_WEST ▪ SOUTH_AMERICA ▪ ASIA ▪ CHINA ▪ ALL.
probeResultType	**Optional.** The type of probe result to view in the report being requested. If not specified, this will be ALL.	▪ ALL ▪ SUCCESS ▪ FAILURE.
reportStartDateTime	**Optional.** The reportStartDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Details Report is being requested. StartDateTime must not be more than three (3) months old. The date range (begin – end date) for the report cannot be greater than seven (7) days. If not specified, this will be set to last twenty four (24) hours.	Date-time.
reportEndDateTime	**Optional.** The reportEndDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Probe Result Details Report is being requested. The date range (begin – end date) cannot be greater than seven (7) days	Date-time.
limit	**Optional.** The number of rows per page for paginated responses. The default is 1,000 if not specified. The maximum value that can be provided for the limit is 1,000.	Integer.

Response: If task completes, Status Code 200 OK is returned with the *Probe Result Details V2 Output DTO* and the following data:

Response Body	Description	Type
sbTcProbeDetailsRow	The list of Probe Result Details.	Array.
sbTcProbeDetailsCount	The number of Rows in the report.	Long.

Errors: An error code is returned under the following conditions:

- If the reportEndDateTime is earlier than the reportStartDateTime.
- If the duration between reportStartDateTime and reportEndDateTime exceeds 7 days.
- If the reportStartDateTime is older than 3 months.
- If the {accountName } cannot be accessed by the current userName.

Probe Results Details v2 Output DTO

Table 207 Probe Result Details v2 Output DTO

Field	Description	Type
accountName	The Account associated to the Probe Result Details row.	String.
zoneName	The Zone associated to the Probe Result Details row.	String.
poolName	The Pool associated to the Probe Result Details row.	String
trafficServicePoolType	The Pool type associated to the Probe Result Details row.	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER
probeResultType	The probeResultType from the request. Value of ALL if both a Success and Failure Probe Result Details Report is being requested.	▪ SUCCESS ▪ FAILURE ▪ ALL
poolRecord	The record within the pool that was probed.	String.
poolRecordType	The type of the record within the pool that was probed.	▪ A ▪ AAAA ▪ CNAME
poolProbeRegion	The region from which the pool record was probed.	▪ North America East ▪ North America West ▪ North America Central ▪ Europe East ▪ Europe West

Field	Description	Type
		▪ South America ▪ Asia ▪ China
probeLogTime	The Date and Time when probing began for the record from the probe region.	Date-Time.
probeResult	The response for the probe that determined the probe result.	String.
probeType	The type of method that was used for probing.	String.
statusCount	The number of times this result was received for this probe request.	Integer.

JSON Example: Probe Result Details Report v2 Response


```
{
  "sbTcProbeDetailsRow": [
    {
      "accountName": "javauie2e",
      "zoneName": "e2e-slb.com.",
      "poolName": "slbpool1.e2e-slb.com.",
      "trafficServicePoolType": "SIMPLE_LOAD_BALANCING",
      "probeResultType": "ALL",
      "probeLogTime": "2016-06-14T00:00:00.000Z",
      "poolRecord": "74.125.138.99",
      "poolRecordType": "A",
      "poolProbeRegion": "US-EAST",
      "probeResult": "Success",
      "probeType": "HTTP",
      "statusCount": 4,
    },
    {
      "accountName": "javauie2e",
      "zoneName": "e2e-slb1.com.",
      "poolName": "slbpool1.e2e-slb1.com.",
      "trafficServicePoolType": "SITE_BACKER",
      "probeResultType": "FAILURE",
      "probeLogTime": "2016-06-14T00:00:00.000Z",
      "poolRecord": "74.125.138.99",
      "poolRecordType": "A",
      "poolProbeRegion": "US-EAST",
      "probeResult": "Success",
      "probeType": "HTTP",
      "statusCount": 4
    }
  ],
  "sbTcProbeDetailsCount": 5479
}
```

Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** **GET** <v2/reports/traffic_services/probe_result/details?accountName=GTV8&cursorOperation=NEXT&limit=1000>; rel="next" **Relative URL to previous page of report if available:** <v2/reports/traffic_services/probe_result/details?accountName=GTV8&cursorOperation=PREVIOUS&limit=1000>; rel="previous"

If a Report's Results exceed 1000 records per page, you can use the "Next / Previous" header command to search the additional results.

Failover Report

The **Failover Report** displays the details for failover and/or failback events that occurred for an account, within a provided time frame as well as the reason the event occurred. When a Failover Report is requested, the report is returned to the user immediately.

Requesting Failover Report

Method and URI:

```
GET https://api.ultradns.com/reports/traffic_services/failover_report
```

Body: None

Parameters: Must include *Failover Query Parameters.*

Failover Query Parameters

Table 208 Failover Query Parameters

Field	Description	Type
accountName	**Required.** The Account for which the Failover Report is being requested	String.
zoneName	**Optional.** The Zone under the account for which the Failover Report is being requested.	String.
poolName	**Optional.** The Simple Load Balancing Pool under the Zone, and under the Account specified for which the Failover Report is being requested.	String.
poolRecord	**Optional.** The record within the pool for which the report is being requested.	String
poolRecordType	**Optional.** The type of record within the pool for which the report is being requested. If not specified, this will be ALL.	▪ A ▪ AAAA ▪ CNAME ▪ ALL
trafficServicePoolType	**Optional.** The type of Pool for which the Failover Report is being requested. If not specified, value will be ALL	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER ▪ ALL
reportStartDateTime	**Optional.** The reportStartDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for	Date-time.

Field	Description	Type
	which the Failover Report is being requested. StartDateTime must not be more than three (3) months old. The date range (begin – end date) for the report cannot be greater than thirty (30) days. If not specified, this will be set to last twenty four (24) hours.	
reportEndDateTime	**Optional.** The reportEndDateTime in ISO 8601 UTC **(yyyy-MM-ddTHH:mm:ss.SSSZ)** for the period for which the Failover Report is being requested. The date range (begin – end date) cannot be greater than thirty (30) days	Date-time.
limit	**Optional.** The number of rows per page for paginated responses. The default is 1,000 if not specified. The maximum value that can be provided for the limit is 1,000.	Integer.

Response: If task completes, Status Code 200 OK is returned with the *Failover Output DTO* and the following data:

Response Body	Description	Type
probeResultDetails	The list of Failover records.	Array.
probeResultDetailsCount	The number of Rows in the report.	Long.

Errors: An error code is returned under the following conditions:

- If the reportEndDateTime is earlier than the reportStartDateTime.
- If the duration between reportStartDateTime and reportEndDateTime exceeds 30 days.
- If the reportStartDateTime is older than 3 months.
- If the {accountName} cannot be accessed by the current userName.

Failover Output DTO

Table 209 Failover Output DTO

Field	Description	Type
accountName	The Account associated to the Failover row.	String.
zoneName	The Zone associated to the Failover row.	String.
poolName	The Pool associated to the Failover row.	String
poolRecord	The record within the pool that was probed.	String
poolRecordType	The type of the record within the pool that was probed.	▪ A ▪ AAAA ▪ CNAME
poolRecordState	The state of the record associate to the Failover row.	▪ ACTIVE ▪ INACTIVE
poolRecordStatus	The status of the pool record associated to the Failover row.	String
failoverReason	The reason of the failover event .	String
allFailRecord	The all fail record indicator.	▪ True ▪ False
trafficServicePoolType	The Pool type associated to the Failover row.	▪ SIMPLE_LOAD_BALANCING ▪ SIMPLE_FAILOVER ▪ TRAFFIC_CONTROLLER ▪ SITE_BACKER
failoverTime	The Date and Time when probing began for the record from the probe region.	Date-Time.

JSON Example: Failover Response

```
{
   "failoverRecordCount": 1000,
   "failoverRecords": [
    {
      "failoverTime": "2020-0929T08:56:14",
      "accountName": "javauie2e",
      "zoneName": "regex.com.",
      "poolName": "200.regex.com.",
      "poolRecord": "75.125.23.76",
      "poolRecordState": "ACTIVE",
      "failoverReason": "Probe Success",
      "allFailRecord": true,
      "poolRecordStatus": "ok",
      "trafficServicePoolType": "SITEBACKED",
```

```
      "poolRecordType": "A",
    },
    {
      "failoverTime": "2020-0929T08:58:14",
      "accountName": "javauie2e",
      "zoneName": "regex.com.",
      "poolName": "201.regex.com.",
      "poolRecord": "74.125.23.76",
      "poolRecordState": "INACTIVE",
      "failoverReason": "Probe Failure",
      "allFailRecord": true,
      "poolRecordStatus": "ok",
      "trafficServicePoolType": "TRAFFIC_CONTROLLER",
      "poolRecordType": "A",
    }
   ]
}
```


Response Link Headers

Field	Description
Link	**Relative URL to next page of report if available:** **GET** <v2/reports/traffic_services/probe_result/ failover_report?accountName=GTV8&cursorOperation=NEXT &limit=1000>; rel="next" **Relative URL to previous page of report if available:** <v2/reports/traffic_services/probe_result/ failover_report?accountName=GTV8& cursorOperation=PREVIOUS&limit=1000>; rel="previous"

If a Report's Results exceed 1000 records per page, you can use the "Next / Previous" header command to search the additional results.

Document Revisions

The following table provides a list of major revisions made to this document since its initial release. Be sure you are using the current version of the document.

Date	Version	Changes
2021-03-31	3.18.0	▪ The *Host Directional Response Counts Report* now contains two new DTO parameters that when used, can provide up to ten million records in .CSV response, as well as providing Source IP details at the host level. The new parameters are **zoneNames** and **wrap**.
2021-02-25	3.16.0	▪ Additional error scenarios were added to User Creation, Account Management, and Security Group Management API calls to reinforce that an (target) Account must be in an **Active** status for a user to be added or moved to an account or group.
2021-01-29	3.14.0	▪ The "Max Number of Days in Past Allowed" (*Reporter Service Report Properties*) value for the **Host Query Volume Report** and the **Host Daily Query Volume Report** has been changed from thirteen (13) months to ninety (90) days. ▪ The **startDate** field for the Host Query Volume Report and Host Daily Query Volume Report has been updated to allow for only a maximum of ninety (90) days in the past (from today's date).
2021-01-08	3.13.0	▪ The **changeComment** field now supports the usage of the colon (:) special character. ▪ When using a colon (:) as a search paramter for the Audit Log Report, users will need to use " ***/:*** " (slash colon) instead of just a colon (:).
2020-12-10	3.12.0	▪ A new field titled **changeComment** has been added to the *Zone Create DTO*. This variable allows users to attach a change comment to zones that can be viewed and searched for in the Audit Log report. New APIs have been added to the following sections in this User Guide: ▪ *Probe Result Summary v2 Report* - This is enhanced version of *Probe Result Summary Report* that can retrieve report with least number of mandatory fields with better performance. ▪ *Probe Result Details v2 Report* - This is enhanced version of *Probe Result Details Report* that can retrieve report with least number of mandatory fields with better performance. ▪ *Failover Report* - This is a new API that displays the details for failover and/or failback events that occurred within a time frame.

Date	Version	Changes
2020-12-04	3.11.0	▪ A new api call for *Export a Zone* has been added. This call allows for a zone, or multile zones, to be exported into a BIND file format.
2020-10-04	3.9.0	▪ *Create Web Forwards* now recognizes the use of the anchor character (#) as a unique identifier. This change will prevent an error from occurring on duplicate records if a matching record utilizes the anchor character (#) in the url. ▪ *Create Zone via BIND File Upload* now supports the usage of whitespace (blank space) for zone names when more than one zone are being created.
2020-08-19	3.3.1	▪ *Synchronous Zone Query Volume Report* is now available in the Reporting APIs section.
2020-08-17	3.3.0	▪ *SiteBacker and Traffic Controller Pool Notifications* are now only able to be created for Primary Records. All Fail Records are no longer supported. ▪ The *Create Multiple RRSets in a Zone via BIND File Upload* call has been updated to indicate that only new records can be created using this method. When the call is used to update existing records, and 202 response will be returned with no action taken. ▪ *Table 9 Primary Zone DTO* – **Inherit** parameter has received an updated description explaining that if zone transfer settings have already been set for the account, the Inherit value will default to ALL.
2020-07-22	3.2.0	▪ Users can now add a Global (Directional) Group for more than one record in the same pool by using the isExistingGroupFromPool parameter. See JSON Example: Add Existing Global Group to a Directional Pool on page *95* of this guide for more details.
2020-07-08	3.1.0	▪ The *Get DNSSEC Details for a Zone* api call now returns the **dnsKeyReocrd** value for zones that are signed using the On_the_Fly singing method. ▪ Updates were made to the *Get DS Record* call, as well as the *Get SSHFP Record* call. Each call now correctly displays the Method and URI to avoid “Data Not Found” errors. o Changed end path parameter from {zoneName} to {ownerName}.
2020-05-20	2.83.1	▪ An enhanced explanation of how SiteBacker and Traffic Controller pool records are returned when there are duplicate priority values assigned to records. See the *Priorities* section for more details.

Date	Version	Changes
2020-04-30	2.83.0	▪ Record Permissions now support the SPF Record Type in the *SecurityGroupEntry DTO* and the *SecurityException DTO*.
2020-04-01	2.82.0	▪ The explanation between the Type and Permission attributes for setting Permissions at the Record and or Pool level has been expanded upon in the *SecurityGroupEntry DTO*.
2020-03-09	2.81.0	▪ Added JSON examples for *Invite New User* for additional clarity of the body content.
2020-01-15	2.79.0	▪ A new parameter called **failureThreshold** has been added to the *SiteBacker Pool Fields*. This new parameter allows you to designate the minimum number of records that must fail for a pool to be labeled FAILED.
2019-10-11	2.72.0	▪ New field parameters have been added to the *SiteBacker Pool Fields* – **backupRecords/backupRecord/availableToServe** , and *RDataInfo Fields* – **status**. These new parameters allow you to check if the pool is active and available to serve records.
2019-09-23	2.71.0	▪ A new field in the *HTTP Probe Details DTO structure* has been added called transactions/expectedResponse, which allows you to specify the expected response codes for a probe.
2019-09-13	2.70.0	▪ A new form of DNSSEC signing is now available called On the Fly signing. With this new signing feature, Advanced Records can be signed as well as Traffic Management Pools. Refer to the *On The Fly Signing* section for more details. ▪ Additional clarification and content has been added for *Response Link Headers* in the Reports section.
2019-08-07	2.67.1	▪ A new report has been added to the REST API User Guide - *Country Code Directional Response Counts Report*.
2019-08-07	2.67.0	▪ Users can now create a “NULL” MX record that will indicate that the record / domain does not accept email. Adding a null MX record will cause all mail delivery attempts to a domain to fail immediately. Refer to the *RRSet DTO* section on how to add a Null MX record.
2019-07-01	2.65.0	▪ When performing the *Get All Exceptions for an Object* call, the **groupName** attribute will now return the username for STANDALONE groups, instead of FirstName LastName. ▪ The Account DTO – *User DTO* now includes the field **authType**, which returns a user’s authentication / login type.

Date	Version	Changes
		▪ The *GET Users of an Account* call can now be returned in a .CSV format.
2019-06-07	2.64.0	▪ The *GET Users of an Account* and *Get Users in a Security Group* will now return the list of user names in ascending order.
2019-05-28	2.63.0	▪ A new field has been added to the Security Preferences DTO called **oldPassword**. This will be used when performing a PUT or PATCH call when updating the password for the account.
2019-05-08	2.62.0	▪ The *GET Users of an Account* api call now returns the field **apiOnlyUser**, which determines whether or not the user has access to the API, or just the UI. ▪ The **Permissions** field in the *SecurityGroupEntry DTO* has been updated to display the new permission types available for each Security Group type. ▪ Provided additional content and screen shots to support using the BIND upload feature to create or update zones.
2019-04-11	2.59.0	▪ A new field type **"relativeForwardType**" has been added to the *WebForward DTO*. This field type allows you to append the target path. ▪ A new field type "**usersCount**" has been added to the *Security Group DTO*. This field returns the total number of users in a group (except for standalone groups) on a GET call.
2019-03-11	2.57.0	▪ The *DS Records* type is now available and supported via the REST API.
2019-02-25	2.56.0	▪ The on-demand *Zone Snapshot and Restore APIs* feature has been added to the REST API. Pelase refer to the Support Page – Zone Snapshot and Restore API Guide for further details on how to use this feature.
2019-01-11	2.54.0	Updates have been made to the following sections in this User Guide: ▪ The *SSHFP Records* type is now available and supported. ▪ The *Transfer Status Details DTO* has been added to the Zone API call, along with the **transferStatusDetails** parameter being added to the *Zone DTO*. ▪ The *UsageLimit DTO* has been added to the Extended Accounts API call, along with the **usageLimit** parameter being added to the *Account DTO*.

Date	Version	Changes
		▪ A new report has been added to the REST API User Guide - *Usage Summary Report*.
2018-12-10	2.53.0	▪ Two new calls have been added in the User Creation section: *Get Pending Invitations* and *Delete Pending Invitations*. ▪ The REST API URIs have been changed (for every API call) to a more streamlined version, that also does not include the usage of the /v1/ or /v2/ parameter when making a call. For more details, refer to the *UltraDNSAPI Versioning* section of this User Guide.
2018-11-26	2.52.0	Updates have been made to the following sections in this User Guide: ▪ *Get Details of Current User / Update Details of Current User* – Updated the JSON example and API call description allowing for updating User details. ▪ *Account DTO* – Account Name Servers parameter has been added, allowing users to retrieve the Active or Pending Name Server details. ▪ *Invite New User / Re-Invite User* – Updated the API call description and details. ▪ *Assign User to a Security Group* – Added a disclaimer about how the API call operates, and the two possible functions for adding new Users to a security group.
2018-11-09	2.51.0	Updates have been made to the following sections in this User Guide: ▪ Two new reports have been added to the REST API User Guide: *Host Directional Response Counts Report* and *Postal Code Directional Response Counts Report*. ▪ The *Get Account Info* details section now contains a GET and UPDATE API call for the **Account Holder Address** details. ▪ You can now retrieve the **Agent Description** when performing a *Get SiteBacker Agents for account*.
2018-10-01	2.49.0	Two new reports have been added to the REST API User Guide: *Client IP Directional Response Counts Report* and *Zone Directional Response Counts Report*.
2018-09-07	2.47.0	A new query parmater **systemGenerated** has been added to the Resource Record Sets that returns whether a record is system generated or not. For additional details, refer *Table 28* and *Table 30*.
2018-08-24	2.46.0	A new field to provide TSIG Algorithms while creating Primary Zones or Secondary Zones via Zone Transfer has been added. Refer to *Table 9* - **nameserver/ tsigAlgorithm**, and *Table 15* -

Date	Version	Changes
		nameServerIpList/nameServerIP1/tsigAlgorithm, **nameServerIpList/nameServerIP2/tsigAlgorithm**, and **nameServerIpList/nameServerIP3/tsigAlgorithm**.
		New IP Probe Region names have been implemented for *SiteBacker and Traffic Controller Pool Probes*. The list of regions has been expanded from four to eight new regions. Refer to *Table 53* for the new region names.
2018-08-09	2.45.0	*Parameters for get metadata for zones* has been updated to include two new parameters – **dnssec_status** and **account_name**. Along with the account_name parameter is a new format to eliminate white space in the account names.
2018-07-27	2.44.0	Added the new **status** parameter to the *Table 46 SiteBacker Pool Fields* which will be returned with all SiteBacker and Traffic Controller GET calls.
2018-07-13	2.43.0	A new report has been added to the **Reporting Section** - *Class C Network Level Directional Response Counts Report*.
		We have streamlined the Response messages for all of the API calls in this guide.
2018-06-28	2.42.0	Added the *Volume Change Report* details to the Reporting Section.
		Updated the *Sitebacker Agent / Probes* to reflect four new IPs, and the removal of two previous IPs.
2018-06-01	2.40.0	An additional explanation about combining GeoIP and Source IP together in a Directional Pool record has been provided in the *Configuring GeoIP and Source IP Together*.
		The *Delete Access of a User from an Account* call has been added which allows you to remove a user's access to an account.
2018-05-29	2.39.0	Added an additional parameter to the *Audit Log Response DTO*, as well as additional API call examples.
		Added a new Report to the Reporting Section of the REST API User Guide: *Host Level Advanced Response Codes*.
2018-04-19	2.37.0	Updated the **JSON Example: User List DTO** under the *Accounts API* with new output details.
		Test Probe has received a new variable called **followDirect** which is used to enable/disable the auto HTTP redirection for test probe
2018-04-04	2.36.0	Included content in the *SiteBacker and Traffic Controller Pool Probes* for changes that are coming soon to the Probe Regions
2018-03-23		Included the section for *TTL Records Consistency in Sitebacker/Traffic Controller Pool Records*.
2018-02-23		A new parameter for *Batch API* and *Batch Query API* called **Async** has been added, which will provide a Task-Id and run as a background task, so that large Batch calls will not timeout and fail.
2018-02-10		Provided an explanation for an error related to too many requests being received from an IP address on the REST API in the *429 Error Response* section.
2018-02-09		Added additional example scenarios to better explain the TTL value consolidation for RRSets (via BIND file uploads or various types of requests).

Date	Version	Changes
2018-01-12		Added additional clarity for the usage of *Making Updates via JSON PATCH Format* as well as provided an example of using JSON PATCH when *Partially Update an Account-level SourceIP Group*.
2017-12-08		Added a new parameter called *ignoreECS* to the *Directional Pools API* section along with new JSON examples.
2017-11-14		Added the new parameter "availableToServe" to *RDataInfo Fields* which will apply to SiteBacker and Traffic Controller pools.
2017-11-03		The *Tasks* section has been updated to provide greater clarity for the usage of the "Q" query parameter.
2017-10-06		The **API only Access** feature added on the UltraDNS Managed Services Portal will now prevent users from using their username and password to access to the UltraDNS Portal. These users will only have access to the REST API. The *Authorization* section has more details.
2017-09-22		Added the *Advanced Response Codes Report* section to the *Reporting* section.
2017-08-11		Added the *Raw Query Sample Report* in the *Reporting* section.
2017-08-10		Added the *TTL Records Consistency in RD Pool Records* section for how the RD pool TTL updates will be handled.
2017-06-06		Added the *Suspend a Zone* and *UnSuspend a Zone* calls in the Zone API section.
2017-05-16		Added a security note for the new Two Factor Mobile Authentication function being applied to the UltraDNS Managed Services Portal.
2017-05-03		Updated the *Sitebacker Agent / Probes* to provide additional IPs for probing.
2017-03-21		Added the new endpoint for *Get SiteBacker Agents for* .
2017-03-07		Updated *Table 12 Secondary Zone DTO* to include the **notificationEmailAddress** DTO, and updated the corresponding JSON Example.
2017-03-01		Updated the Directional Pool API – Profile section to include the new output parameter: **rdatainfo/type** which will return the record type for the pool or subpool. Applicable GET JSON examples have been updated witth the new output parameter. Refer to the *Sitebacker Agent / Probes* section for additional reference(s).
2017-02-17		Updated the *SecurityPreferences DTO* to include the extended password length (now 36 characters long) and the additional Special Characters now supported.
2017-02-13		Updated the *Web Forwards* section to include more detailed JSON examples, and DTO updates.
2017-02-08		Updated the *Security Preferences* section, and the DTOs associated.
2017-01-12		Included documentation for the addition of the *Batch Query API* function, as well as updates to support IPv6 in Simple Monitor / Failover pools, and *Zone DNSSEC APIs* updates.
2017-01-09		Updated the Extended Accounts API – Account DTO section to include the return of "features" per account, with JSON examples.

Date	Version	Changes
2017-01-03		Updated *Table 3 JSON PATCH DTO* to include the “move” operation. Updated Directional Pool TTLs with a more in-depth description of functionality along with a step by step display of how TTLs are displayed via pool / subpool / record level.