

Table of contents

[INTRODUCTION](#)[CLIENT LIBRARIES](#)[ACCOUNT](#) +[ZONES & RECORDS](#) +[DATA SOURCES & FEEDS](#) +[FILTER CHAIN™](#) +[MONITORING & NOTIFICATIONS](#) +[REPORTING & STATISTICS](#) +[PULSAR](#) +[DHCP](#) +[IPAM](#) +[DYNAMIC DNS](#) +

Introduction

This is the NS1 API (v1) documentation which provides you with the necessary tools for managing zone, records, data sources and feeds, account settings, and any other aspects of your NS1 account. The NS1 API is a standard REST API with JSON responses. The HTTP method you use — **GET**, **PUT**, **POST**, or **DELETE** — determines the type of action to be taken by the API. Generally, in the case of **PUT** and **POST** requests, you will send a JSON body in the request with details about the resource you're creating or updating.

Most API requests are authenticated with a simple API Key, which you must specify in the **X-NSONE-Key** request header. You can manage your API Keys in the NS1 portal ([my.nstone.net](#)).

Note: We highly recommend that you treat your NS1 API keys as secrets and that you configure the API key with only the minimal access needed for your application. Refer to [this Help Center article](#) for details.

Refer to the [NS1 Help Center](#) for our complete library of technical documentation. If you have any questions or experience any issues while using the API, please [submit a support ticket](#).

Table of contents

INTRODUCTION

CLIENT LIBRARIES

ACCOUNT ▾

General +

User +

API key ▾

GET List all API keysGET View API key detailsPUT Create an API keyPOST Edit an API keyDELETE Delete an API key

Team +

IP "allow" list +

Billing +

Payment method +

Activity Log +

GET /v1/account/apikeys/<keyID>

View API key details

Returns details, including **permissions**, for the NS1 API key indicated by its API key ID. **Note:** The API key ID is not the same as the key secret.

 Copy Code

Returns details about the specified API key (by key ID)

```
curl -X GET -H "X-NSONE-Key: $NSONE_API_KEY" https://api.nzone.net/v1/apikeys/<keyID>
```

 Expand Code Examples

User +

API key -

[GET](#) List all API keys[GET](#) View API key details[PUT](#) Create an API key[POST](#) Edit an API key[DELETE](#) Delete an API key

Team +

IP "allow" list +

Billing +

Payment method +

Activity Log +

ZONES & RECORDS +

DATA SOURCES & FEEDS +

FILTER CHAIN™ +

MONITORING & NOTIFICATIONS +

REPORTING & STATISTICS +

PULSAR +

Warning: NS1 recommends restricting access to your API key and reducing permissions only to include what the application requires. Keep your API key secret. If it is compromised, delete the API key immediately. Restrict access for this API key by applying IP addresses or CIDR blocks in the `ip_whitelist` property (ex. `ip_whitelist: ['104.20.48.182', '104.20.49.0/24']`).

name <i>string</i>	(Required) Name for this API key. NS1 recommends naming the API key something that aligns with the application for which it is being used.
teams <i>array of strings</i>	Array of team IDs corresponding to teams with which to associate this API key. If assigned to a team, the API key inherits the permissions set for that team.
ip_whitelist_strict <i>boolean</i>	Set to "true" to restrict access to only those IP addresses and networks listed in the <code>ip_whitelist</code> field.
ip_whitelist <i>array of strings</i>	A list of IP addresses or subnets to which to grant the API key access. Optionally, set the <code>ip_whitelist_strict</code> field to "true" to restrict access to <i>only</i> the IP addresses and subnets listed.
permissions <i>object</i>	Set of supported permissions to enable or disable for this API key, including: <ul style="list-style-type: none">• account (<i>object</i>) - Set of account-related permissions, including:<ul style="list-style-type: none">• manage_account_settings (<i>boolean</i>) - Set to "true" to allow the API key to manage account-related settings.• manage_teams (<i>boolean</i>) - Set to "true" to allow the API key to create or modify teams.



CLIENT LIBRARIES

ACCOUNT +

ZONES & RECORDS +

DATA SOURCES & FEEDS -

Data Sources +

Data Feeds -

GET View active data feeds for a source**GET** View data feed details**PUT** Connect a new data feed to a data source**POST** Modify a data feed**DELETE** Disconnect a data feed**POST** Publish data from a data source

FILTER CHAIN™ +

MONITORING & NOTIFICATIONS +

REPORTING & STATISTICS +

PULSAR +

Returns the details of a single data feed, including **config** details and any **record**, **region**, or **answer** metadata table **destinations**.

Path parameters:

Parameter	Description
sourceid <i>string</i>	(Required) The ID of the data source that contains the data feed that you want to review. You can retrieve this ID by viewing your active data sources .
feedid <i>string</i>	(Required) The ID of the data feed that you want to review. You can retrieve this value by viewing active data feeds for your specified sourceid .

Request URL:

```
https://api.nsonone.net/v1/data/feeds/<sourceid>/<feedid>
```



FILTER CHAIN™

Filters

- Cost
- Geofence Country
- Geofence Regional
- Geotarget Country
- Geotarget Latlong
- Geotarget Regional
- Netfence ASN
- Netfence Prefix
- Priority
- Pulsar Sort
- Pulsar Stabilize
- Select First N
- Select First Region
- Shed Load
- Shuffle
- Sticky Shuffle
- Sticky Region
- Up

Filters

To create any filter via the API, you must create or update a record for your domain with the answers you plan to use. You must also define any metadata that would be required for the filters you want to create. Finally, you specify the resource record to which to apply these changes. You create via a **PUT** request, and you update via a **POST** request.

Example

Suppose that you want to create an A record with two answers on the **pulsar.example.com** domain that uses the Pulsar Performance Sort filter. In the first step of this process, we often recommend that you start by defining backup logic, such as basic geographical steering, in order to have backup logic if Pulsar does not have enough data to make a decision. To do this, you would:

1. Define the **zone**, **domain**, and the record **type**.
2. Define at least one **answer** within the record.
3. Add geographic metadata using the **meta** object for the answer to define the backup filter.

So far, your command would read as follows:

[DATA SOURCES & FEEDS](#) +[FILTER CHAIN™](#) +[MONITORING & NOTIFICATIONS](#) +[REPORTING & STATISTICS](#) -[QPS](#) +[Usage](#) -[GET](#) View account-wide usage statistics[GET](#) View zone-level usage statistics[GET](#) View record-level usage statistics[Lease stats](#) +[Scheduled Reports](#) +[PULSAR](#) +[DHCP](#) +[IPAM](#) +[DYNAMIC DNS](#) +[GET](#) `/v1/stats/usage/:zone`

View zone-level usage statistics

Please refer to `/stats/usage` for a detailed explanation of the output and query parameters. Note that `by_tier` (`true/false`) parameters are no longer supported.

When querying usage stats, note the following:

- If **none** of the fields are present, the response includes aggregate stats for the entire account.
- If only the **zone** field is present, the response includes aggregate stats for the specified zone.
- If **domain**, **rectype**, and **zone** fields are present in the object, the response includes stats for the specified record.

Parameter	Type	Description	Required?
zone	string	Name of the zone	yes
period	enum(string)	Relative time. Possible values: '1h', '24h', '30d' (Default: '24h')	no



FILTER CHAIN™ +

MONITORING & NOTIFICATIONS +

REPORTING & STATISTICS +

PULSAR +

DHCP +

IPAM -

Networks -

[GET](#) View a list of networks[GET](#) View network details[GET](#) View a network report[PUT](#) Create a network[POST](#) Update a network[DELETE](#) Remove a network

Subnets +

Pools +

Metadata tags +

DYNAMIC DNS +

[POST](#) /v1/ipam/network/:networkID

Update a network

Updates an existing IPAM network specified in the URL path by its unique `networkID`.

Path parameters:

Parameter	Type	Description
networkID	integer	Required. Unique identifier for the network.

Request body parameters:

Parameter	Type	Description
name	string	Required. A unique name for the IPAM network.

