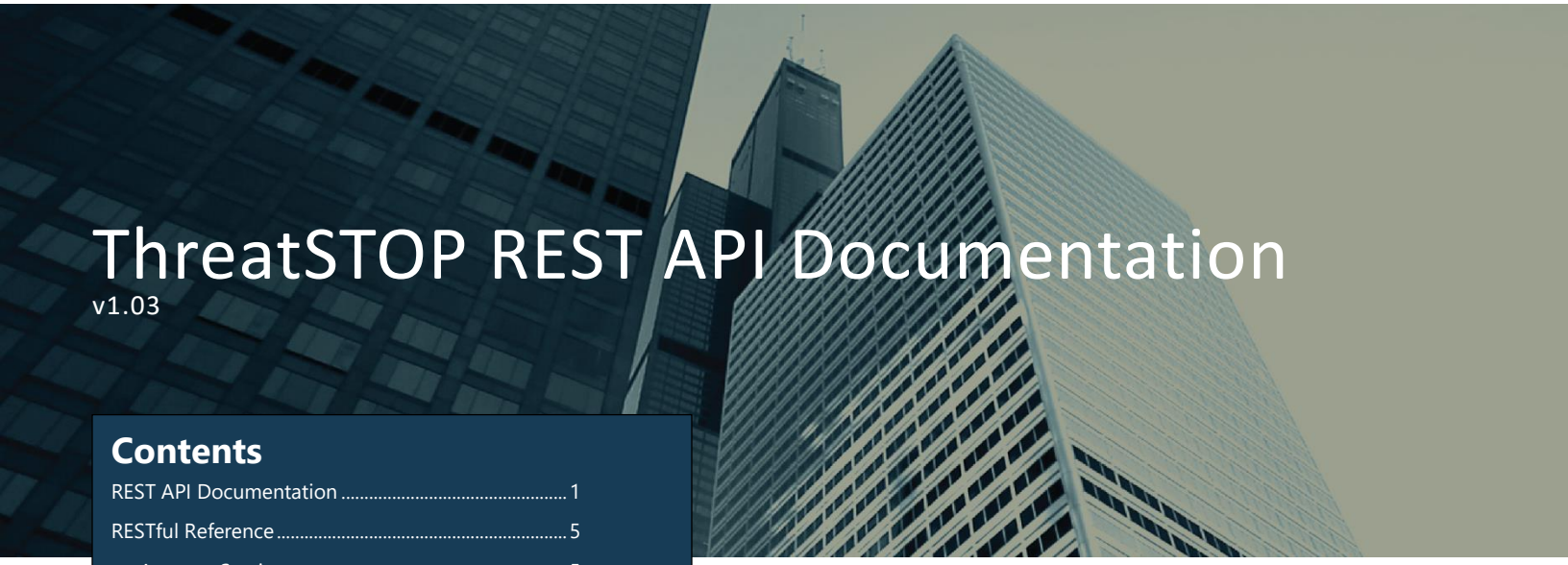
ThreatSTOP

www.threatstop.com



ThreatSTOP REST API Documentation

v1.03

Contents

Data provided by ThreatSTOP is intended for business use. A licensed, active account and API key will be required to access data and interface with our systems.

REST API Documentation

This document covers the ThreatSTOP REST API, and is intended for integration of ThreatSTOP's system by software developers - for example:

- Account and device provisioning
- Policy and User-defined lists customization
- Integration with a Security Information Event Management (SIEM/SEM)

Data contained in this document is subject to change and expansion.

This manual is laid out as a reference manual comprised of two sections:

1. An overview of the REST API, including formats and settings required to access it.
2. A reference manual of API calls

Currently, the following API services and their sub-services are documented:

- Accounts
  - accounts
    - access
- Check Indicator of Compromise
  - check_ioc
- Devices
  - devices
- Logs
  - logs
- Policy Services
  - domain_policies
  - ip_policies
- User Lists Services
  - user_ip_lists
  - user_domain_lists

The documentation itself is not intended to be a complete primer on API programming but should be sufficient enough for a user with limited programming experience to be able to program a simple interface with the system.

As such the introductory section of the document covers what RESTful Programming is, why it is useful, and how it is used. When it is used is outside the scope of this document and is not covered.

Getting Started

Requirements

Accessing the ThreatSTOP API requires an active account and API key.

API Key Generation

A unique API key is generated and provided on all ThreatSTOP accounts. The key will appear on the **Home** page of your ThreatSTOP account at the bottom of the page under **API Keys**.

Why REST?

A completely in-depth primer on REST can be found at the Learn REST: A RESTful Tutorial site (http://www.restapitutorial.com/). While this documentation does contain some information about REST, its primary focus is on the use of RESTful standards with the ThreatSTOP REST API.

What is REST?

REST, meaning REpresentational State Transfer, is the architecture that defines the basic functions of retrieving data from a server. By defining constraints on how a client can request and handle data, it creates the possibility of transferring data in an easy to handle format.

How is REST Used?

The REST API is accessed using any programming language that supports the HTTPS protocol. Modern languages will provide high-level libraries to format requests and process responses. Many stand-alone tools are available as well, such as:

- Unix Command line
  - https://curl.haxx.se/
  - https://github.com/jkbrzt/httpie
- Windows
  - Windows PowerShell
- Web browser plug-ins
  - Postman

Note:

Examples throughout this text will use cURL commands to provide reference.

HTTP Verbs

Interfacing with the ThreatSTOP REST API centers around the use of *HTTP Verbs*: **GET**, **PUT**, **POST**, **PATCH,** and **DELETE**. Here we're going to be a little more technical in their function and provide examples of the format of each verb.

GET

GET requests are used to request data from a service. This can be returned in a number of formats, but under REST is generally returned as JSON or XML. ThreatSTOP's service will always return data to the client in JSON format. GET content is idempotent, making the same call will return the same value.

POST

POST requests are used to create new resources. A definition of the new object is provided in JSON or XML.

PUT

PUT requests are intended to replace an entire resource. This is effectively an overwrite command. The command itself is entirely idempotent, in that if the content is created with PUT, and then called again the data will be placed on the server with the same state.

PATCH

PATCH requests are used to update specific fields of a resource in place. That is, where POST creates an entirely new resource, and PUT overwrites an existing resource, PATCH will change the content of an existing resource without overwriting or recreating it.

DELETE

The easiest to understand, DELETE removes data associated with a URI. On the server side, this will either fully remove the data or produce an error if the resource doesn't exist.

	POST	GET	PUT	PATCH	DELETE
/collection	Create New Object	List/Search	Not supported	Not supported	Not supported
/collection/uuid	Not supported	Read Object	Update (replace) Object	Update (partial) Object	Delete object

Not all verbs apply to every API service. Please check the reference of a service for the specific commands that it supports.

JSON Format

The JavaScript Object Notation (JSON) format, is intended to provide a data exchange format that is easy for humans to read and follow, while also being easily processed by computers. The format, while inspired by JavaScript, is language agnostic and imports into any programming language with little overhead for developers.

The structure itself is based around name value pairs and ordered list values.

More information about JSON may be found from Introducing JSON (http://json.org/).

URL Format

Requesting data with a RESTful API involves calling the server for the API, calling the API function desired, and passing in relevant data for the request and to authenticate the user's right to access the system. For the ThreatSTOP API the base of the call will take the following format:

```
https://rest.threatstop.com/v4.0/<resource>
```

Caution:

As iterations of the API are released, the version number will change. Existing programs or scripts that access the REST API will have a grace period to update to before a non-supported version of the API is sunset.

Additionally, it should be noted that both the JSON and URL Formats are case sensitive.

Versioning

Versioning is currently limited to v4.0, and is called in the URL as denoted in *URL Format* above.

Minor version updates will be backward compatible.

HTTP Codes

The ThreatSTOP API replies with HTTP codes according to REST best practices. It also includes detailed error messages in the body of responses to requests that triggered an error. See *HTTP Status and Error Codes* for a reference of HTTP and Application error codes.

- 2xx codes represent a successful request.
- 4xx codes represent a problem with the request.
- 5xx codes represent a problem with the API service. Please contact support@threatstop.com if a 5xx error persists.

Example Error Response:

```
{
    "additional_info": {
        "detail": "<error description>",
        "error_code": <error code>
    },
    "error_description": "<response
description>",
    "status_code": <http response code>
}
```

Universally Unique Identifiers (UUID)

Resources are uniquely identified by a UUID, presented in v4 format. More information about may be found here:

- *RFC 4122*, IETF (https://tools.ietf.org/html/rfc4122)
- *UUID Version 4 (random)*, Wikipedia, (https://en.wikipedia.org/wiki/Universally_unique_identifier#Version_4_.28random.29)

Authentication

In order to use ThreatSTOP's REST API, an API key associated with your account must be provided. All requests will require that the API key be provided. It is transmitted in the HTTP 'Authorization' header.

Example:

```
curl -H 'Authorization: <your api key>` https://rest.threatstop.com/v4.0/<resource>
```

Objects

The following Objects are provided by the API:

- **_data**: All objects created or retrieved will be listed in this object.
- **_link**: All objects will also return a _link which can be used to identify related resources. The links available depend on the service being accessed.

RESTful Reference

This section covers API interfaces, and is separated subsystem.

Account Services

Account services covers account administration, this is specifically covered by:

- **accounts** – which deals with the creation, viewing, and maintenance of users.
  - **accounts access** this is a sub-service of the account API which can be used to grant third party access to an account.

accounts specifically uses GET, POST, and PUT to retrieve information or update and maintain information about a user account. Specifically by calling the API with the format:

```
https://rest.threatstop.com/v4.0/<resource>
```

In the case of accounts the call would look like:

```
https://rest.threatstop.com/v4.0/accounts/<account_object_id>
```

Accessing the sub-service of accounts (**accounts_access**) requires calling the **accounts** service, providing the account **object_id**, then providing the sub-service to be used (**accounts_access**):

```
https://rest.threatstop.com/v4.0/accounts/<account_object_id>/accounts_access/<object_id>
```

accounts

GET

Returns account information for the requested account. Additionally, the *show_api_tokens* query parameter will return the API tokens associated with the account.

URL:	https://rest.threatstop.com/v4.0/accounts/
Required query parameters:	None
Optional query parameters:	show_api_tokens (GET)
Accept:	n/a
Request Headers Required:	Yes

Response Status Codes

Response statuses are provided based on transaction requests

Successful:	200
Device not found:	400
Invalid API token:	401
Object not found:	404

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests with a bad object_id:	10300
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100

Response Body

For successful requests, we will return a 200 status code with a list of account(s).

- If no **account_object_id** is specified, the list will consist of all accounts to which the user of the auth token has access.
- If an **account_object_id** is specified, the list will consist of a single account object (the one specified).

```
{
   "_data": [{
     "object_id": "<object_id>",
     "email": "a<last name>@threatstop.com",
     "salutation": "<title up to seven characters>",
     "first_name": "<first name>",
     "last_name": "test",
     "company_name": "<company name>",
     "phone_number": "555-555-5555",
     "address1": "555 5th ave",
     "address2": "apt 5",
     "city": "san diego",
     "state": "CA",
     "postal_code": "92115"
     "country": "United States",
     "website": "<company website>",
     "active": true,
     "_links": {
       "access": {
         "href": "http://rest.threatstop.com/v4.0/accounts/<object_id>/access"
       },
       "self": {
         "href": "http://rest.threatstop.com/v4.0/accounts/<object_id>"
       }
     }
   }],
   "_links": {
     "self": {
       "href": "http://rest.threatstop.com/v4.0/accounts"
     }
   }
 }
```

POST

This will create a new account.

URL:	https://rest.threatstop.com/v4.0/accounts
Required query parameters:	email, password, first_name, last_name, account_type
Optional query parameters:	None
Accept:	application/json
Request Headers Required:	Yes

Request Body

```
{
  "email": "<last name>@threatstop.com",
  "password": "<password>",
  "salutation": "<title up to seven characters>",
  "first_name": "<first name>",
  "last_name": "<last name>",
  "account_type": "<account type>",
  "company_name": "<company name>",
  "phone_number": "555-555-5555",
  "address1": "555 5th ave",
  "address2": "apt 5",
  "city": "san diego",
  "state": "CA",
  "postal_code": "92115",
  "country": "United States",
  "website": "<company website>",
  "active": true,
  "agree_to_terms_and_conditions": true
}
```

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	400

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with invalid parameters:	13002
For requests with a reserved (black listed) email username:	16000
For requests with a reserved (black listed) email domain:	16001

Response Body

For successful requests the following JSON object is returned:

```
{
  "_data": [{
    "object_id": "<object id>",
    "email": "a<last name>@threatstop.com",
    "salutation": "<title up to seven characters>",
    "first_name": "<first name>",
    "last_name": "test",
    "company_name": "<company name>",
    "phone_number": "555-555-5555",
    "address1": "555 5th ave",
    "address2": "apt 5",
    "city": "san diego",
    "state": "CA",
    "postal_code": "92115"
    "country": "United States",
    "website": "<company website>",
    "active": true,
    "_links": {
      "access": {
        "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access"
      },
      "self": {
        "href": "http://rest.threatstop.com/v4.0/accounts/<object id>"
      }
    }
  }],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/accounts"
    }
  }
}
```

PUT

This will update an existing account.

URL:	https://rest.threatstop.com/v4.0/accounts/<account_object_id>
Required query parameters:	password, first_name, last_name, account_type
Optional query parameters:	None
Accept:	application/json
Request Headers Required:	Yes

Request Body

```
{
  "password": "<password>",
  "salutation": "<title up to seven characters>",
  "first_name": "<first name>",
  "last_name": "<last name>",
  "account_type": "<account type>",
  "company_name": "<company name>",
  "phone_number": "555-555-5555",
  "address1": "555 5th ave",
  "address2": "apt 5",
  "city": "san diego",
  "state": "CA",
  "postal_code": "92115",
  "country": "United States",
  "website": "<company website>",
  "active": true,
  "agree_to_terms_and_conditions": true
}
```

Note:
PUT is a *replace into* operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Authorization Error:	403
Device not found:	404

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests with a bad object_id:	10300
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with invalid parameters:	13002

Response Body

For successful requests:

```
{
  "_data": [{
    "object_id": "<object id>",
    "email": "a<last name>@threatstop.com",
    "salutation": "<title up to seven characters>",
    "first_name": "<first name>",
    "last_name": "test",
    "company_name": "<company name>",
    "phone_number": "555-555-5555",
    "address1": "555 5th ave",
    "address2": "apt 5",
    "city": "san diego",
    "state": "CA",
    "postal_code": "92115"
    "country": "United States",
    "website": "<company website>",
    "active": true,
    "_links": {
      "access": {
        "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access"
      },
      "self": {
        "href": "http://rest.threatstop.com/v4.0/accounts/<object id>"
      }
    }
  }],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/accounts"
    }
  }
}
```

access

GET

Return a list of accounts granted 3rd party access to the given account.

URL:	https://rest.threatstop.com/v4.0/accounts/<account_object_id>/access/<access_object_id>
Required query parameters:	None
Optional query parameters:	None
Accept:	n/a
Request Headers Required:	Yes

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Device not found:	400
Invalid API token:	401
Authorization Error:	403
Object not found:	404

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100

Response Body

For successful requests, a 200 status code is returned along with account details.

- If no **access_object_id** is specified, the list will consist of all accounts to which the user of the auth token has access.
- If an **access_object_id** is specified, the list will consist of a single account object (the one specified).

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/"
    },
    "account": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>"
    }
  },
  "_data": {
    "third_party_access_to_this_account": [{
      "access_object_id": "<access object id>",
      "email": "<email address for administrator>",
      "access_level": "full",
      "links": {
        "self": {
          "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<access
object id>"
        }
      }
    }],
    "accounts_this_account_can_manage": [{
      "object_id": "<object id>",
      "email": "reporting@threatstop.com",
      "access_level": "reporting",
      "_links": {
        "self": {
          "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<object
id>"
        }
      }
    }]
  }
}
```

POST

This will grant access to another account

URL:	https://rest.threatstop.com/v4.0/accounts/<account_object_id>/access
Required query parameters:	third_party_access
Optional query parameters:	None
Accept:	application/json
Request Headers Required:	Yes

Request Body

```
{
  "third_party_access": [{
    "email": "<email address>",
    "access_level": "full"
  }, {
    "email": "reporter@threatstop.com",
    "access_level": "reporting"
  }]
}
```

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Device not found:	400
Invalid API token:	401
Authorization Error:	403

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400

Response Body

For successful requests:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<access object
id>"
    },
    "account": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>"
    }
  },
  "_data": {
    "third_party_access_to_this_account": [{
      "access_object_id": "<access object id>",
      "email": "<email address for administrator>",
      "access_level": "full",
      "links": {
        "self": {
          "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<access
object id>"
        }
      }
    }],
    "accounts_this_account_can_manage": []
  }
}
```

PUT

This will update an access object for a given account

URL:	https://rest.threatstop.com/v4.0/accounts/<account_object_id>/access/<access_object_id>
Required query parameters:	email, access level
Optional query parameters:	None
Accept:	application/json
Request Headers Required:	Yes

Request Body

```
{
  "email": "<email address for administrator>",
  "access_level": "full"
}
```

Note:
PUT is a *replace into* operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Device not found:	400
Invalid API token:	401
Authorization Error:	403
Device not found:	404

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests with a bad object_id:	10300
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400

Response Body

For successful requests:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<access object
id>"
    },
    "account": {
      "href": "http://rest.threatstop.com/v4.0/accounts/<object id>"
    }
  },
  "_data": {
    "third_party_access_to_this_account": [{
      "access_object_id": "<access object id>",
      "email": "<email address for administrator>",
      "access_level": "full",
      "links": {
        "self": {
          "href": "http://rest.threatstop.com/v4.0/accounts/<object id>/access/<access
object id>"
        }
      }
    }],
    "accounts_this_account_can_manage": []
  }
}
```

DELETE

URL:	https://rest.threatstop.com/v4.0/accounts/<account_object_id>/access/<access_object_id>
Required query parameters:	None

This will delete the requested access object.

Optional query parameters:	None
Accept:	n/a
Request Headers Required:	Yes

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Authorization Error:	403
Device Not Found:	404

Error Conditions

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests with a bad object_id:	10300
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_links": {
    "self": {
      "href": "http://localhost:5000/v4.0/accounts/<account_object_id>/access"
    },
    "account": {
      "href": "http://localhost:5000/v4.0/accounts/<account_object_id>"
    }
  },
  "_data": {
    "accounts_this_account_can_manage": [],
    "third_party_access_to_this_account": []
  }
}
```

Valid Parameters - Account Services

Field	Type	Restrictions	Acceptable Values
email	string	3-64 characters, rfc5322	n/a
password	string	8-32 characters	n/a
salutation	string	0-7 characters	n/a
first_name	string	1-25 characters	n/a
last_name	string	1-25 characters	n/a
account_type	string	1-25 characters	a10*
address1	string	0-255 characters	n/a
address2	string	0-255 characters	n/a
city	string	0-60 characters	n/a
state	string	0-60 characters	see state section below
postal_code	string	0-25 characters	n/a
country	string	0-60 charactters	see country section below
website	string	0-255 characters	n/a
active	boolean	n/a	true, false
agree_to_terms_and_conditions	boolean	n/a	true, false

Possible values include **Trial**, **Community**, and **RPZ**.

CheckIOC Service

An Indicator of Compromise (IOC) can consist of an IP address, Domain, or a domain with at least one subdomain and a leading wild card (for example, *.google.com). The IOC service returns data grouped into two or three categories depending on the IOC being an IP address or a Domain. Domain IOCs will return all three values, while IP Address IOCs will not return **Related Records**. The categories break down as follows:

- **Active Records**: Returns a list of active records for the given IOC. These records include the IOC date identified, last seen, and a list of the targets in which the IOC is located.
- **Historic Records**: Returns a list of historic records for the given IOC. These records include the IOC date identified, last seen, and a list of the targets in which the IOC is located.
- **Related Records**: If the IOC is a domain, this will return a list of related IP addressed (A records) for the domain.

GET

This will return data for the supplied IOCs.

URL:	https://rest.threatstop.com/v4.0/check_ioc
Required Query Parameters:	ioc
Optional Query Parameters:	None
Accept:	n/a
Request Headers Required:	Authorization: <API key>

Response Status Codes

Invalid API token:	401
Successful:	200
Bad request:	400

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with a bad IP address:	15000
For requests with a bad domain address:	15001
For requests without the required parameter:	13002

Response Body

For successful requests, a 200 status code is returned, along with the data for the requested IOC.


```
{
    "_data":[
        {
            "info":{
                "related_records":[

                ],
                "active":[
                    {
                        "blockers":[
                            {
                                "last_update":1476367599,
                                "danger_level":"3",
                                "name":"DPHISH",
                                "description":"Domains
 used in phishing attacks. This list may contain false positives as
phishing pages are frequently located on compromised but otherwise
legitimate websites",
                                "short_description":"PHISHING DOMAINS",
                                "public_description":"Domains
 used in phishing attacks. This list may contain false positives as
phishing pages are frequently located on compromised but otherwise
legitimate websites"
                            }
                        ],
                        "ioc":"app.adjust.com",
                        "first_identified":1476281258,
                        "last_used":1476367599,
                        "domain":"app.adjust.com"
                    },
                    {
                        "blockers":[
                            {
                                "last_update":1476367525,
                                "danger_level":"4",
                                "name":"DPHISHTA",
                                "description":"PhishTank phishing Domains",
                                "short_description":"PhishTank phishing Domains",
                                "public_description":"Domains that are used in current
phishing attacks from phishtank.com."
                            }
                        ],
                        "ioc":"app.adjust.com",
                        "first_identified":1476281185,
                        "last_used":1476367525,
                        "domain":"app.adjust.com"
                    }
                ],
                "history":[

                ]
            },
            "ioc":"app.adjust.com"
        }
    ],
    "_links":{
        "self":{
            "href":"http://rest.threatstop.test/v4.0/check_ioc?ioc=app.adjust.com"
        }
    }
}
```

POST

This will return data for the supplied IOCs

URL:	https://rest.threatstop.com/v4.0/check_ioc
Required Query Parameters:	None
Optional Query Parameters:	None
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body Parameters

Required Body Parameters:	ioc
Optional Body Parameters:	last_seen strategy targets

The IOCs parameter takes a list of objects. In each object:

- **ioc** - is required it and can be a domain or IP address.
- **last_seen** - optional. Number of seconds from now to look back (for example, 1 month would be 2592000).
- **strategy** - optional. "include" or "exclude." Does nothing if not used with targets.
- **targets** - optional. a comma separated list of targets (for example, BOTSBLK, TS-RANS, MSISACEX, etc.) to either include or exclude.

Request Body

```
{
    "iocs":[
        {
            "ioc":"app.adjust.com",
            "strategy":"exclude",
            "targets":[
                "DPHISH"
            ],
            "last_seen":2592000
        },
        {
            "ioc":"192.0.2.1",
            "strategy":"include",
            "targets":[
                "TS-RANS"
            ],
            "last_seen":25920000
        }
    ]
}
```

Note:

The targets are referenced using their programmatic names, and not the clear text names.

Response Status Codes

Successful:	200
Invalid API token:	401
Bad request:	400

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with a bad ip address:	15000
For requests with a bad domain address:	15001
For requests without the required parameter:	13002

Response Body

For successful requests, we will return a 200 status code with a data for the given IOCs.


```
{
    "_data":[
        {
            "info":{
                "related_records":[

                ],
                "active":[
                    {
                        "blockers":[
                            {
                                "last_update":1476367525,
                                "danger_level":"4",
                                "name":"DPHISHTA",
                                "description":"PhishTank phishing Domains",
                                "short_description":"PhishTank phishing Domains",
                                "public_description":"Domains that are used in current
phishing attacks from phishtank.com."
                            }
                        ],
                        "ioc":"app.adjust.com",
                        "first_identified":1476281185,
                        "last_used":1476367525,
                        "domain":"app.adjust.com"
                    }
                ],
                "history":[

                ]
            },
            "ioc":"app.adjust.com"
        },
        {
            "info":{
                "active":[
                    {
                        "blockers":[
                            {
                                "last_update":1476367651,
                                "danger_level":"5",
                                "name":"TS-RANS",
                                "description":"Addresses that the ThreatSTOP security
team has determined are current and active ransomware C&C or distribution sites",
                                "short_description":"TSCritical Ransomware IP Addresses",
                                "public_description":"Addresses that the ThreatSTOP
security team has determined are current and active ransomware C&C or distribution sites"
                            }
                        ],
                        "ioc":"192.0.2.1",
                        "address":"192.0.2.1",
                        "last_used":1476367651,
                        "first_identified":1475187841
                    }
                ],
                "history":[

                ]
            },
            "ioc":"192.0.2.1"
        }
    ],
    "_links":{
        "self":{
            "href":"http://rest.threatstop.test/v4.0/check_ioc"
        }
    }
}
```

Valid Parameters - Check IOC

Field	Type	Restrictions	Acceptable Values
iocs*	array	A list of IOCs, maxiumum of 50 elements. IOC is required.	n/a
items	object	The IOC, can be an IP address, or a domain name.	e.g., '192.0.2.1' example.com
last_seen	number	IOC was last seen less than (seconds) ago.	n/a
ioc	string	The IOC, can be an IP address, or a domain name.	e.g., '192.0.2.1' example.com
targets	array	Look at IOCs that match these targets. Returns values in a string format.	n/a

* **iocs** is required, with **ioc** being required within it.

Device Services

Devices covers device creation and removal. Its use requires that each request include an API token that is sent in the request header.

devices specifically uses GET, POST, PUT, DELETE to retrieve information, update and maintain information about the designated device, or remove the device from the account. Specifically by calling the API with the format:

```
https://rest.threatstop.com/v4.0/<resource>
```

In the case of devices the call would look like:

```
https://rest.threatstop.com/v4.0/devices/<user_device_id>
```

All requests validate that an API token is present and is valid. If one is not provided, the API will return a 401 error. We then will validate that the required fields are present and return a 400 if they are not present. We then will check if the data passed is valid and return a 400 if it is not. If the request passes the previous checks, we will do the action indicated by HTTP method.

GET

This will return the requested device.

URL:	https://rest.threatstop.com/v4.0/devices/<user_device_id>
Required query parameters:	None
Optional query parameters:	None
Accept:	n/a

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Device not found:	400

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests where a device_id is specified, but a device is not found:	10300

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
    }
  },
  "_data": [{
    "object_id": "<object_id>"
    "ip_address": "192.0.2.0",
    "device_manufacturer": "<Manufacturer>",
    "device_model": "<Device Type>",
    "device_nickname": "test",
    "serial_number": "12345",
    "service_type": "Trial",
    "tdid": "tdid_e7fa64d3",
    "device_class": "Class C",
    "policy": {
      "object_id": "<object_id>"
      "name": "TSAll-CNEE",
      "_links": {
        "policy": {
          "href": "http://rest.threatstop.com/v4.0/policies/<policy_id>"
        }
      }
    },
    "_links": {
      "self": {
        "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
      }
    }
  }]
}
```

POST

This will create a new device.

URL:	https://rest.threatstop.com/v4.0/devices
Required query parameters:	ip_address, service_type, device_class
Optional query parameters:	None
Accept:	application/json

Request Body

```
{
 "device_nickname": "",
  "service_type": "",
  "device_manufacturer": "",
  "device_model": "",
  "device_class": "",
  "serial_number": "",
  "ip_address": "",
}
```

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	201
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an ip in use:	12001
For requests where the user exceeds his device limit:	12002
For requests with an invalid policy:	12003
For requests that specify a device_id:	10301

Response Body

For successful requests an HTTP response of 201 (Successful) is returned along with the JSON object:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
    }
  },
  "_data": [{
    "object_id": "<object_id>"
    "ip_address": "192.0.2.0",
    "device_manufacturer": "<device_manufacturer>",
    "device_model": "<device_model>",
    "device_nickname": "test",
    "serial_number": "12345",
    "service_type": "Trial",
    "tdid": "<tdid_number>",
    "device_class": "Class C",
    "policy": {
      "object_id": "<object_id>"
      "name": "TSAll-CNEE",

      "_links": {
        "policy": {
          "href": "http://rest.threatstop.com/v4.0/policies/<policy_id>"
        }
      }
    },
    "_links": {

      "self": {
        "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
      }
    }
  }]
}
```

PUT

This will create a new device.

URL:	https://rest.threatstop.com/v4.0/devices/user_device_id
Required query parameters:	ip_address, service_type, device_class
Optional query parameters:	None
Accept:	application/json

Request Body

```
{
  "device_nickname": "",
  "service_type": "",
  "device_manufacturer": "",
  "device_model": "",
  "device_class": "",
  "serial_number": "",
  "ip_address": "",
}
```

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests where no device_id is specified:	10006
For requests with an ip in use:	12001
For requests with an invalid policy name:	12003
For requests where a device_id is specified, but a device is not found:	10300
For requests where the device name is already being used:	12005

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
    }

  },
  "_data": [{
    "object_id": "<object_id>"
    "ip_address": "192.0.2.0",
    "device_manufacturer": "<device_manufacturer>",
    "device_model": "<device_model>",
    "device_nickname": "test",
    "serial_number": "12345",
    "service_type": "Trial",
    "tdid": "<tdid_number>",
    "device_class": "Class C",
    "policy": {
      "object_id": "<object_id>"
      "name": "TSAll-CNEE",

      "_links": {
        "policy": {
          "href": "http://rest.threatstop.com/v4.0/policies/<policy_id>"
        }
      }
    },
    "_links": {

      "self": {
        "href": "http://rest.threatstop.com/v4.0/devices/<user_device_id>"
      }
    }
  }]
}
```

DELETE

This will delete the requested device.

URL:	https://rest.threatstop.com/v4.0/devices/<user_device_id>
Required query parameters:	None
Optional query parameters:	None
Accept:	n/a

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests where the device is not found:	10300
For requests where you try to delete the collection:	12004

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_links": {
    "self": {
      "href": "http://rest.threatstop.com/v4.0/devices"
    }
  },
  "_data": []
}
```

Valid Parameters - Devices

Field	Type	Restrictions	Acceptable Values
device_nickname	string	max length 10 characters	n/a
service_type	enum	n/a	["Trial", "Community", "Paid"]
device_manufacturer	string	n/a	n/a
device_model	string	n/a	n/a
device_class	enum	n/a	["Class A", "Class B", "Class C"]
serial_number	string	n/a	n/a
ip_address	string	user_ip_list: IP or IP Range **Note**: The IP address provided should be the Public IP address associated with the account, as defined in the Introduction to the user Portal. **Caution**: IPs in the following ranges are forbidden: ['10.0.0.0/8', '127.0.0.0/8', '172.16.0.0/12', '192.168.0.0/16', '169.254.0.0/16', '0.0.0.0/8']	e.g., '192.0.2.1' '192.0.2.0-192.0.2.255'
location	string	n/a	n/a
postal_code	string	n/a	n/a
policy	string	n/a	n/a

Logs Service

The Logs service allows for the creation of a Log Upload and retrieval method completely unique to your environment.

Policy services uses GET, POST, PUT, DELETE to retrieve information, update and maintain information about the designated logs, or remove the policy from the account. By calling the API with the format:

```
https://rest.threatstop.com/v4.0/<resource>
```

In the case of the Logs Service the call would look like:

```
https://rest.threatstop.com/v4.0/logs/<object_id>
```

All requests confirm that an API token is present and is valid. If one is not provided, the API will return a 401 error. We then will validate that the required fields are present and return a 400 if they are not. We then check if the data passed is valid and return a 400 if it is not. If the request passes the previous checks, we will perform the action indicated by HTTP method.

GET

This will return the requested account.

URL:	https://rest.threatstop.com/v4.0/logs/<object_id>
Required Query Parameters:	None

Optional Query Parameters:	• **Paging**: These optional query parameters are used for paging. These only apply to requests for the collection (https://rest.threatstop.com/v4.0/logs). If an object_id is specified, these will be ignored. For the most part, users will not be using these directly, they'll be using one of the links in a response (self, next, previous)
	  ○ limit: number of logs to return, default is 10, max 100.
	  ○ index: object_id of a log to start at. Used in conjunction with direction.
	  ○ direction: "next" or "previous" are acceptable values. Defaults to next if not specified. "Next" gives the next set of logs that older than the index. Previous gives logs newer than the index.
	• **Filters:** These optional query parameters are used for filtering. These only apply to requests for the collection (https://rest.threatstop.com/v4.0/logs). If an object_id is specified, these will be ignored.
	  ○ device: device object_id to filter logs by. If this is specified, only logs that will be returned are from that device.

Accept:	N/A
Request Headers Required:	Authorization: <auth key>

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Device not found:	400
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests with a bad object_id:	10300

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

- If no **object_id** is specified, the list will consist of the number of logs specified by the limit (default 10) to which the user of the auth token has access.
- If an **object_id** is specified, the list will consist of a single log object (the one specified).


```
{
 "_links": {
  "self": {
   "href":
"http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=next"
  },
  "next": {
   "href":
"http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=next"
  },
  "previous": {
   "href":
"http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=previous"
  }
 },
 "_data": [
  {
   "blocked_in": 0,
   "date_received": 1481914374,
   "errors": 0,
   "blocked_out": 2149,
   "allowed_in": 0,
   "date_last_entry": 1481914373,
   "date_processed": 1481914377,
   "allowed_out": 701,
   "skipped": 0,
   "date_first_entry": 1481910773,
   "object_id": "<object_id>",
   "matches": 10000,
   "error_code": "0",
   "status": "imported formatted files",
   "device": {
    "object_id": "<object_id>",
    "_links": {
     "self": {
      "href": http://rest.threatstop.test/v4.0/devices/<object_id>
     }
    }
   },
   "_links": {
    "self": {
     "href": http://rest.threatstop.test/v4.0/logs/<object_id>
    }
   }
  }
 ]
}
```

Field info

Field	Notes
blocked_in	These are updated as the log progresses through our system
blocked_out	
allowed_in	
allowed_out	
matches	
date_received	All dates are epochs. These are updated as log progresses through our system.
date_last_entry	
date_processed	
error	Is only shown if the user is an admin
status	This is field varies based on the user's access level. If the user does not have admin rights, they will see: • received • processing • completed • error Admin privileged users will see: • received • OK • created formatted files • imported formatted files • device parser doesn't exist • error

Field mapping

```
status_lookup = {
  "error": "error",
  "created formatted files": "processing", # finished logs process normalized
  "imported formatted files": "completed", # finished logs process formatted
  "device parser doesn't exist": "error", # no format specified for process normalized
  "format not specified": "error", # no format specified for process normalized
  "ok": "processing", # finished logs parse
  "received": "received"
}
```

URL:	https://rest.threatstop.com/v4.0/logs

POST

This will upload a new log.

Required Query Parameters:	None
Optional Query Parameters:	None
Accept:	multipart/form-data
Request Headers Required:	Authorization: <auth key>

Request Body Parameters

The request body is of a multipart/form-data. Most tools and libraries will take care of the formatting for you. There are two parameters that must be provided.

Required Body Parameters:	• **device**: a device object_id or a tdid • **log**: a log file
Optional Body Parameters:	

Parameter definitions:

- See *Valid Parameters - Reports* section below

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	201
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
No log attached:	23000
Log larger than 15 MB:	23001

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
 "_links": {
  "self": {
   "href":
http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=next
  },
  "next": {
   "href":
"http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=next"


  },
  "previous": {
   "href": http://rest.threatstop.test/v4.0/logs?index=<object_id>&limit=10&direction=previous
  }
 },
 "_data": [
  {
   "blocked_in": 0,
   "date_received": 1481914374,
   "errors": 0,
   "blocked_out": 2149,
   "allowed_in": 0,
   "date_last_entry": 1481914373,
   "date_processed": 1481914377,
   "allowed_out": 701,
   "skipped": 0,
   "date_first_entry": 1481910773,
   "object_id": "<object_id>",
   "matches": 10000,
   "error_code": "0",
   "status": "received",
   "device": {
    "object_id": "<object_id>",
    "_links": {
     "self": {
      "href": http://rest.threatstop.test/v4.0/devices/<object_id>
     }
    }
   },
   "_links": {
    "self": {
     "href": http://rest.threatstop.test/v4.0/logs/<object_id>
    }
   }
  }
 ]
}
```

Field Info

Field	Notes
blocked_in	These are updated as the log progresses through our system.
blocked_out	
allowed_in	
allowed_out	
matches	
date_received	All dates are epochs. These are updated as log progresses through our system.
date_last_entry	
date_processed	
error	Only shown if the user is an admin.
status	This is field varies based on the user's access level. If the user does not have admin rights, they will see: • received • processing • completed • error Admin privileged users will see: • received • ok • created formatted files • imported formatted files • device parser doesn't exist • error

Field Mapping

```
status_lookup = {
  "error": "error",
  "created formatted files": "processing", # finished logs process normalized
  "imported formatted files": "completed", # finished logs process formatted
  "device parser doesn't exist": "error", # no format specified for process normalized
  "format not specified": "error", # no format specified for process normalized
  "ok": "processing", # finished logs parse
  "received": "received"
}
```

Policy Services

Two different services handle Policy creation. The Domain Policy service, and IP Policy service. Both policies allow the programmatic definition of device security policy. This is the confluence of both ThreatSTOP native threat intelligence lists and User-Defined lists--created in the user portal, or with the User List services.

Policy services uses GET, POST, PUT, DELETE to retrieve information, update and maintain information about the designated policy, or remove the policy from the account. By calling the API with the format:

```
https://rest.threatstop.com/v4.0/<resource>
```

In the case of Policy Services the call would look like:

```
https://rest.threatstop.com/v4.0/<domain_policies or ip_policies>/<policy_object_id>
```

All requests confirm that an API token is present and is valid. If one is not provided, the API will return a 401 error. We then will validate that the required fields are present and return a 400 if they are not. We then check if the data passed is valid and return a 400 if it is not. If the request passes the previous checks, we will perform the action indicated by HTTP method.

domain_policies

User Domain Lists are lists that contain domains and can only be added to DNS Policies.

GET

This will return the requested Policy.

URL:	https://rest.threatstop.com/v4.0/domain_policies/<policy_object_id>
Required Query Parameters:	None
Optional Query Parameters:	object_id
Accept:	N/A
Request Headers Required:	Authorization: <API Key>

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Device not found:	400
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests with a bad object_id:	10300

Response Body

For successful requests, we return a 200 status code with a list of Policies.

- If no **policy_object_id** is specified, the list will consist of all Policies that the user of the Authorization: <API Key> token has access to.
- If a **policy_object_id** is specified, the list will consist of a single Policy object (the one specified).


```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<policy>.<account_id>.rpz.threatstop.local",
      "description": "Test Description",
      "global": false,
      "user_lists": [
        {
          "_links": {
            "user_list": {
              "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
            }
          },
          "list_name": "UDD",
          "behavior": "NXDOMAIN",
          "object_id": "<object_id>"
        }
      ],
      "targets": [
        {
          "danger_level": "4",
          "behavior": "NXDOMAIN",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "object_id": "<object_id>",
      "expert_mode": "no",
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/policies"
    }
  }
}
```

POST

This will create a new Policy.

URL:	https://rest.threatstop.com/v4.0/domain_policies
Required Query Parameters:	None
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization: <API Key>

Request Body Parameters

Required Parameters:	policy_name
Optional Parameters:	policy_type dns_name description expert_mode visible public targets* users_lists**

Request Body

```
{
    "policy_name": "Test",
    "targets": [
        {
            "object_id":
"<object_id>",
            "behavior":"NXDOMAIN"
        }
    ],
    "user_lists": [
        {
            "object_id":
"<object_id>",
            "behavior": "block"
        }
    ]
}
```

Parameter definitions:

- See the *Valid Parameters - Policy Services* section below

Note:

* **targets** is required if either **target.object_id** or **target.behavior** is defined in the request.

** **users_lists** is required if either **user_list.object_id** or **user_list.behavior** is defined in the request.

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	201
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with an object_id specified for POST:	10301
For requests with conflicting policy types set:	10302
For requests where the Policy name is already in use:	21000
For requests without policy type set:	21002
For requests that specify expert mode illegally:	21001
For requests with bad target object_id:	10300
For requests with bad target behavior:	21003
For requests with bad user list behavior:	21004

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:


```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<policy>.<account_id>.threatstop.local",
      "description": "",
      "object_id": "<object_id>",
      "expert_mode": false,
      "global": false,
      "targets": [
        {
          "danger_level": "4",
          "behavior": "NXDOMAIN",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "user_lists": [
        {
          "_links": {
            "user_list": {
              "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object_id>"
            }
          },
          "list_name": "Block0",
          "behavior": "block",
          "object_id": "<object_id>"
        }
      ],
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/domain_policies/"
    }
  }
}
```

PUT

This will update an existing Policy. Put is a ***replace into*** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/domain_policies/<policy_object_id>
Required Query Parameters:	object_id
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization: <API Key>

Request Body Parameters

Required Parameters:	policy_name
Optional Parameters:	policy_type dns_name description expert_mode visible public targets* user_lists**

Note:

* **targets** is required if either **target.object_id** or **target.behavior** is defined in the request.

users_lists is required if either **user_list.object_id** or **user_list.behavior** is defined in the request.

Request Body

```
{
    "policy_name": "Test",
    "targets": [
        {
            "object_id":
"<object_id>",
            "behavior":"NODATA"
        }
    ]
}
```

Parameter definitions:

- See *Valid Parameters - Policy Services* section below.

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with conflicting policy types set:	10302
For requests where the Policy name is already in use:	21000
For requests without policy type set:	21002
For requests that specify expert mode illegally:	21001
For requests with bad target object_id:	10300
For requests with bad target behavior:	21003
For requests with bad user list behavior:	21004

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:


```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<policy>.<account_id>.threatstop.local",
      "description": "",
      "object_id": "<object_id>",
      "expert_mode": false,
      "global": false,
      "targets": [
        {
          "danger_level": "4",
          "behavior": "NODATA",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "user_lists": [],
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/domain_policies/<object_id>"
    }
  }
}
```

DELETE

This will delete an existing Policy.

URL:	https://rest.threatstop.com/v4.0/domain_policies/<policy_object_id>
Required Query Parameters:	object_id
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization: <API Key>

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Bad Request:	400
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_data": [],
  "_links": {
    "policies": {
      "href": "http://rest.threatstop.test/v4.0/policies"
    }
  }
}
```

ip_policies

User IP Lists are lists containing IP addresses that can be set to either block or allow and can be added to both IP and DNS Policies.

> **Note:**
>
> IP addresses can be added to both IP and DNS policies. It is important to remember that IP addresses added to DNS policies cannot block inbound communications, only outbound.

GET

This will return the requested Policy.

URL:	https://rest.threatstop.com/v4.0/ip_policies/<policy_object_id>
Required Query Parameters:	None
Optional Query Parameters:	object_id
Accept:	N/A
Request Headers Required:	Authorization

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Device not found:	400
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests with a bad object_id:	10300

Response Body

For successful requests, we will return a 200 status code with a list of Policies.

- If no **policy_object_id** is specified, the list will consist of all Policies to which the user of the auth token has access.
- If a **policy_object_id** is specified, the list will consist of a single Policy object (the one specified).


```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<domain>.<account_id>.rpz.threatstop.local",
      "description": "Test Description",
      "global": false,
      "user_lists": [
        {
          "_links": {
            "user_list": {
              "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
            }
          },
          "list_name": "UDD",
          "behavior": "block",
          "object_id": "<object_id>"
        }
      ],
      "targets": [
        {
          "danger_level": "4",
          "behavior": "block",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "object_id": "<object_id>",
      "expert_mode": "no",
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/policies"
    }
  }
}
```

POST

This will create a new Policy.

URL:	https://rest.threatstop.com/v4.0/ip_policies
Required Query Parameters:	None
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization

Request Body Parameters

Required Parameters:	policy_name
Optional Parameters:	policy_type dns_name description expert_mode visible public targets* user_lists**

Note:

*__targets__ is required if either **target.object_id** or **target.behavior** is defined in the request.

__users_lists__ is required if either **user_list.object_id or **user_list.behavior** is defined in the request.

Request Body

```
{
    "policy_name": "Test",
    "targets": [
        {
            "object_id":
"<object_id>",
            "behavior":"block"
        }
    ],
    "user_lists": [
        {
            "object_id":
"<object_id>",
            "behavior": "block"
        }
    ]
}
```

Parameter definitions:

- See *Valid Parameters - Policy Services* section below.

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	201
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with an object_id specified for POST:	10301
For requests with conflicting policy types set:	10302
For requests where the Policy name is already in use:	21000
For requests without policy type set:	21002
For requests that specify expert mode illegally:	21001
For requests with bad target object_id:	10300
For requests with bad target behavior:	21003
For requests with bad user list behavior:	21004

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:


```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<policy>.<account_id>.threatstop.local",
      "description": "",
      "object_id": "<object_id>",
      "expert_mode": false,
      "global": false,
      "targets": [
        {
          "danger_level": "4",
          "behavior": "block",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "user_lists": [
        {
          "_links": {
            "user_list": {
              "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object_id>"
            }
          },
          "list_name": "Block0",
          "behavior": "block",
          "object_id": "<object_id>"
        }
      ],
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/ip_policies/"
    }
  }
}
```

PUT

This will update an existing Policy. Put is a ***replace into*** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/ip_policies/<policy_object_id>
Required Query Parameters:	object_id
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization

Request Body Parameters

Required Parameters:	policy_name
Optional Parameters:	policy_type dns_name description expert_mode visible public targets* user_lists**

Note:

*__targets__ is required if either **target.object_id** or **target.behavior** is defined in the request.

__users_lists__ is required if either **user_list.object_id or **user_list.behavior** is defined in the request.

Request Body:

```
{
    "policy_name": "Test",
    "targets": [
        {
            "object_id":
"<object_id>",
            "behavior":"block"
        }
    ]
}
```

Parameter definitions:

- See *Valid Parameters - Policy Services* section below

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with conflicting policy types set:	10302
For requests where the Policy name is already in use:	21000
For requests without policy type set:	21002
For requests that specify expert mode illegally:	21001
For requests with bad target object_id:	10300
For requests with bad target behavior:	21003
For requests with bad user list behavior:	21004

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_data": [
    {
      "policy_name": "Test",
      "dns_name": "<policy>.<account_id>.threatstop.local",
      "description": "",
      "object_id": "<object_id>",
      "expert_mode": false,
      "global": false,
      "targets": [
        {
          "danger_level": "4",
          "behavior": "block",
          "object_id": "<object_id>",
          "handle_name": "ADVBLK",
          "_links": {
            "target": {
              "href": "http://rest.threatstop.test/v4.0/targets/<object_id>"
            }
          }
        }
      ],
      "user_lists": [],
      "_links": {
        "self": {
          "href": "http://rest.threatstop.test/v4.0/policies/<object_id>"
        }
      }
    }
  ],
  "_links": {
    "self": {
      "href": "http://rest.threatstop.test/v4.0/ip_policies/<object_id>"
    }
  }
}
```

DELETE

This will delete an existing Policy.

URL:	https://rest.threatstop.com/v4.0/ip_policies/<policy_object_id>
Required Query Parameters:	object_id
Optional Query Parameters:	None
Accept:	application/json
Request Headers Required:	Authorization

Response Status Codes

Response statuses are provided based on transaction requests. GET can return four status codes:

Successful:	200
Bad Request:	400
Invalid API token:	401

Error Conditions

For invalid requests, we return using the standard error handler response body with the relevant information. We use the error codes listed below

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
  "_data": [],
  "_links": {
    "policies": {
      "href": "http://rest.threatstop.test/v4.0/policies"
    }
  }
}
```

Valid Parameters - Policy Services

Field	Type	Restrictions	Acceptable Values
policy_name	string	Alphanumeric characters and dashes, max 255 characters	
dns_name	string	domain characters, max 255 characters	
description	string	max 100 characters	
expert_mode	boolean		
visible	boolean		
public	boolean		
targets	list		list of objects containing object_id and behavior
targets.object_id	string		
targets.behavior	string	max 255 characters	['block', 'allow'] for many targets, custom behaviors/actions for targets that support DNS/RPZ
user_lists	list		list of objects containing object_id and behavior
user_lists.object_id	string		
user_lists.behavior	string	max 255 characters	['block', 'allow'] for User IP Lists, custom behaviors/actions for User Domain Lists.

User Lists Services

User Lists services covers user created target list administration, this is specifically handled through two services:

- **User IP Lists Service** – used to create, and manage custom target lists of IP addresses created by users.
- **User Domain Lists Service** – used to create, and manage custom target lists of domains (URIs) created by users.

Both sub-services GET, POST, PUT, DELETE, and PATCH to retrieve information, update and maintain information about the designated user list, or remove the user list from the account. Specifically by calling the API with the format:

```
https://rest.threatstop.com/v4.0/<resource>
```

In the case of user lists the call would look like:

```
https://rest.threatstop.com/v4.0/<user_ip_lists or
user_domain_lists>/<user_list_object_id>
```

Each individual call will specify the exact format required to access the service and use the corresponding REST verb.

All requests validate that an API token is present and is valid. If one is not provided, the API will return a 401 error. We then will validate that the required fields are present and return a 400 if they are not present. We then will check if the data passed is valid and return a 400 if it is not. If the request passes the previous checks, we will do the action indicated by HTTP method.

User IP Lists Service

The User IP Lists Service is used to create and manage custom IP target lists.

GET

This will return the requested User Lists.

URL:	https://rest.threatstop.com/v4.0/user_ip_lists/<user_list_object_id>
Required query parameters:	None
Optional query parameters:	object_id
Accept:	n/a
Required Request Headers:	Authorization: <API Key>

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Device not found:	400

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests with a bad object_id:	10300

Response Body

For successful requests, we will return a 200 status code with a list of User Lists.

- If no **user_list_object_id** is specified, the list will consist of all User Lists of which the user of the auth token has access.
  For User List Collections, the addresses list is not returned, but its **record_count** and **address_count** fields are.
- If a **user_list_object_id** is specified, the list will consist of a single User List object (the one specified).

```
{
 "_data": [
  {
   "shared": false,
   "_meta": {
    "addresses": {
     "address_count": 1,
     "record_count": 1
    }
   },
   "object_id": "<object id>",
   "description": "",
   "list_name": "List0",
   "list_type": "block",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
    }
   }
  }
 ],
 "_meta": {
  "count": 1
 },
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_ip_lists"
  }
 }
}
```

POST

This will create a new User List.

URL:	https://rest.threatstop.com/v4.0/user_ip_lists
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

```
{
 "list_name": "GoodList",
 "list_type": "allow",
 "addresses": [
  {
   "value": "8.8.8.8",
   "comments": "Google DNS"
  }
 ]
}
```

Parameter Definitions

- See *Valid Parameters – User Lists* section below

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	201
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include an address in forbidden networks:	19012

Response Body

For successful requests an HTTP response of 201 (Successful) is returned along with the JSON object:


```
{
 "_data": [
  {
   "shared": false,
   "_meta": {
    "addresses": {
     "address_count": 1,
     "record_count": 1
    }
   },
   "object_id": "<object id>",
   "addresses": [
    {
     "value": "221.222.222.244/32",
     "address_type": "netmask",
     "comments": ""
    }
   ],
   "description": "",
   "list_name": "List0",
   "list_type": "block",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
    }
   }
  }
 ],
 "_meta": {
  "count": 1
 },
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
  }
 }
}
```

PUT

This will update an existing User List. Put is a **replace into** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/user_ip_lists/<user_list_object_id>
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

- See *Valid Parameters – User Lists* section below

```
{
 "list_name": "BadList",
 "list_type": "block",
 "addresses": [
  {
   "value": "192.0.2.3",
   "comments": "Evil Server",
  }
 ]
}
```

Parameter Definitions

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include an address in forbidden networks:	19012

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
  }
 },
 "_data": [
  {
   "shared": false,
   "list_name": "BadList",
   "addresses": [
    {
     "comments": "Evil Server",
     "value": "192.0.2.3",
     "address_type": "ip"
    }
   ],
   "description": "",
   "list_type": "block",
   "object_id": "<object id>",
   "_links": {
    "self": {
     "href":
"http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
    }
   }
  }

 ]
}
```

DELETE

This will delete an existing User List.

URL:	https://rest.threatstop.com/v4.0/user_ip_lists/<user_list_object_id>
Required query parameters:	None
Optional query parameters:	None
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Parameter Definitions

- none

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Bad Request:	400

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
 "_links": {
  "user_ip_lists": {
   "href": "http://rest.threatstop.test/v4.0/user_ip_lists"
  }
 },
 "_data": []
}
```

PATCH

This will update an existing User List. Put is an ***update into*** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/user_ip_lists/<user_list_object_id>
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

```
{
 "list_name": "BadList",
 "list_type": "block",
 "addresses": [
  {
   "value": "192.0.2.3",
   "comments": "Evil Server",
   "action": "add"
  }
 ]
}
```

Parameter Definitions

- See *Valid Parameters – User Lists* section below

Response Status Codes

Response statuses are provided based on transaction requests.

Invalid API token:	401
Successful:	200

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include an address in forbidden networks:	19012

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
  }
 },
 "_data": [
  {
   "shared": false,
   "list_name": "BadList",
   "addresses": [
    {
     "comments": "Evil Server",
     "value": "192.0.2.3"
    },
    {
     "comments": "test3",
     "value": "8.8.8.9",
     "address_type": "ip"
    },
    {
     "comments": "test4",
     "value": "8.8.8.10",
     "address_type": "ip"
    }
   ],
   "description": "",
   "list_type": "block",
   "object_id": "<object id>",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_ip_lists/<object id>"
    }
   }
  }
 ]
}
```

User Domain Lists Service

GET

This will return the requested User Lists.

URL:	https://rest.threatstop.com/v4.0/user_domain_lists/<user_list_object_id>
Required query parameters:	None
Optional query parameters:	object_id
Accept:	n/a
Required Request Headers:	Authorization: <API Key>

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Device not found:	400

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	10100
For requests with a bad object_id:	10300

Response Body

For successful requests, we will return a 200 status code with a list of User List(s).

- If no **user_list_object_id** is specified, the list will consist of all User Lists to which the user of the auth token has access. For User List Collections, the addresses list is not returned, but its **record_count** and **address_count** fields are.
- If a **user_list_object_id** is specified, the list will consist of a single User List object (the one specified).


```
{
 "_data": [
  {
   "shared": false,
   "_meta": {
    "addresses": {
     "address_count": 1,
     "record_count": 1
    }
   },
   "object_id": "<object id>",
   "description": "",
   "list_name": "List0",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object id>"
    }
   }
  }
 ],
 "_meta": {
  "count": 1
 },
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_domain_lists"
  }
 }
}
```

POST

This will create a new User List.

URL:	https://rest.threatstop.com/v4.0/user_domain_lists
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

```
{
 "list_name": "GoodList",
 "addresses": [
  {
   "value": "example.com",
   "comments": "Example"
  }
 ]
}
```

Parameter Definitions

- See *Valid Parameters – User Lists* section below

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	201
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include address in forbidden networks:	19012

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:

```
{
 "_data": [
  {
   "shared": false,
   "_meta": {
    "addresses": {
     "address_count": 1,
     "record_count": 1
    }
   },
   "object_id": "<object_id>",
   "addresses": [
    {
     "value": "example.com",
     "comments": "Example"
    }
   ],
   "description": "",
   "list_name": "GoodList",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
    }
   }
  }
 ],
 "_meta": {
  "count": 1 },
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
  }
 }
}
```

PUT

This will update an existing User List. Put is a **replace into** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/user_domain_lists/<user_list_object_id>
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

```
{
 "list_name": "BadList",
 "addresses": [
  {
   "value": "evil.threatstop.com",
   "comments": "Evil Server",
  }
 ]
}
```

Parameter Definitions

- See *Valid Parameters – User Lists* section below

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401

Error Condition

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include address in forbidden networks:	19012

Response Body

For successful requests an HTTP response of 200 (Successful) is returned along with the JSON object:


```
{
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object id>"
  }
 },
 "_data": [
  {
   "shared": false,
   "list_name": "BadList",
   "addresses": [
    {
     "value": "evil.threatstop.com",
     "comments": "Evil Server"
    }
   ],
   "description": "",
   "list_type": "block",
   "object_id": "<object id>",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object id>"
    }
   }
  }
 ]
}
```

DELETE

This will delete an existing User List.

URL:	https://rest.threatstop.com/v4.0/user_domain_lists/<user_list_object_id>
Required query parameters:	None
Optional query parameters:	None
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Parameter Definitions

- none

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401
Bad Request:	400

Response Body

For successful requests, an HTTP response of 200 (Successful) is returned along with the JSON object.

```
{
 "_links": {
  "user_domain_lists": {
   "href": "http://rest.threatstop.test/v4.0/user_domain_lists"
  }
 },
 "_data": []
}
```

PATCH

This will update an existing User List. Put is an **update into** operation. Any optional parameters not sent with the request will be replaced by their defaults (in most cases empty strings).

URL:	https://rest.threatstop.com/v4.0/user_domain_lists/<user_list_object_id>
Required query parameters:	list_name, list_type
Optional query parameters:	description, addresses, addresses.value, addresses.comments
Accept:	application/json
Required Request Headers:	Authorization: <API Key>

Request Body

```
{
 "list_name": "BadList",
 "list_type": "block",
 "addresses": [
  {
   "value": "bad.threatstop.com",
   "comments": "Sinister",
   "action": "add"
  }
 ]
}
```

Parameter Definitions

- See *Valid Parameters – User Lists* section below.

Response Status Codes

Response statuses are provided based on transaction requests.

Successful:	200
Invalid API token:	401

Error Codes

For invalid requests, a standard error handler response body is returned with the relevant information, including one of the following error codes:

Error Condition	Error Code
For requests without an Auth Token:	11000
For requests with an invalid Auth Token:	11001
For requests with an expired Auth Token:	11002
For requests with bad parameters:	11400
For requests with an object_id specified for POST:	10301
For requests where the User List name is already in use:	19000
For requests with an invalid address type:	19050
For requests that exceed the max address range limit:	19011
For requests with domains that exceed the max character count:	19013
For requests with domain segments that exceed the max character count:	19014
For requests that include an address in forbidden networks:	19012

Response Body

For successful requests, an HTTP response of 200 (Successful) is returned along with the JSON object.


```
{
 "_links": {
  "self": {
   "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
  }
 },
 "_data": [
  {
   "shared": false,
   "list_name": "BadList",
   "addresses": [
    {
     "comments": "Evil Server",
     "value": "1.1.1.1"
    },
    {
     "comments": "test3",
     "value": "8.8.8.9"
    }
   ],
   "description": "",
   "list_type": "block",
   "object_id": "<object_id>",
   "_links": {
    "self": {
     "href": "http://rest.threatstop.test/v4.0/user_domain_lists/<object_id>"
    }
   }
  }
 ]
}
```

Valid Parameters - User Lists

Field	Type	Restrictions	Acceptable Values
object_id	string	max 37 characters	n/a
list_type	string	only for User IP Lists	['block', 'allow']
list_name	string	max 8 characters	n/a
shared	boolean	n/a	n/a
description	string	max 1024 characters	n/a
addresses	list		list of objects containing a value, comments, and/or action properties
addresses.value	string	user_ip_list: IP, IP Range up_domain_list: FQDN **Caution**: IP's in the following ranges are forbidden: ['10.0.0.0/8', '127.0.0.0/8', '172.16.0.0/12', '192.168.0.0/16', '169.254.0.0/16', '0.0.0.0/8']	eg. '1.1.1.1' '1.1.1.1-1.1.1.255' 'example.com'
addresses.comments	string	max 50 characters	n/a
addresses.action	string	PATCH only	['add', 'remove']

Appendix

Code Parameters

The following are valid parameters to pass to our API in your code:

Field	Type	Restrictions	Acceptable Values
email	string	3-64 characters, rfc5322	n/a
password	string	8-32 characters	n/a
salutation	string	0-7 characters	n/a
first_name	string	1-25 characters	n/a
last_name	string	1-25 characters	n/a
account_type	string	1-25 characters	a10
address1	string	0-255 characters	n/a
address2	string	0-255 characters	n/a
city	string	0-60 characters	n/a
state	string	0-60 characters	See country section below
postal_code	string	0-25 characters	n/a
country	string	0-60 characters	See country section below
website	string	0-255 characters	n/a
active	boolean	n/a	true, false
agree_to_terms_and_conditions	boolean	n/a	true, false

* this is configurable. For A10, it's only A10. Other possible values include Trial, Community, and RPZ.

State

Alberta
Alabama
Alaska
American Samoa
Arizona
Arkansas
British Columbia
California
Colorado
Connecticut
Delaware
District of Columbia
Fed. States of Micronesia
Florida
Georgia
Guam
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Manitoba
Marshall Islands
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
Newfoundland
New Brunswick
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Northwest Territories
Northern Mariana Islands
Nova Scotia
Ohio
Oklahoma
Ontario
Oregon
Palau
Pennsylvania
Prince Edward Island
Puerto Rico
Rhode Island
Quebec
South Carolina
South Dakota
Saskatchewan
Tennessee
Texas
Utah
Vermont
Virgin Islands
Virginia
Washington
West Virginia
Wisconsin
Wyoming
Yukon Territory

Country

United States
United Kingdom
Afghanistan
Albania
Algeria
American Samoa
Andorra
Angola
Anguilla
Antarctica
Antigua and Barbuda
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bermuda
Bhutan
Bolivia
Bosnia and Herzegovina
Botswana
Bouvet Island
Brazil
British Indian Ocean Territory
Brunei Darussalam
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada
Canadian Nunavut Territory
Cape Verde
Cayman Islands
Central African Republic
Chad
Chile
China
Christmas Island
Cocos (Keeling Islands)
Colombia
Comoros
Congo
Cook Islands
Costa Rica
Cote D'Ivoire (Ivory Coast)
Croatia (Hrvatska)
Cuba
Cyprus
Czech Republic
Denmark
Djibouti
Dominica
Dominican Republic
East Timor
Ecuador
Egypt
El Salvador
Equatorial Guinea
Eritrea
Estonia
Ethiopia
Falkland Islands (Malvinas)
Faroe Islands
Fiji
Finland
France
France, Metropolitan
French Guiana
French Polynesia
French Southern Territories
Gabon
Gambia
Georgia
Germany
Ghana
Gibraltar
Greece
Greenland
Grenada
Guadeloupe
Guam
Guatemala
Guinea
Guinea-Bissau
Guyana
Haiti
Heard and McDonald Islands
Honduras
Hong Kong
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kiribati
Korea (North)
Korea (South)
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Macau
Macedonia
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Marshall Islands
Martinique
Mauritania
Mauritius
Mayotte
Mexico
Micronesia
Moldova
Monaco
Mongolia
Montserrat
Morocco
Mozambique
Myanmar
Namibia
Nauru
Nepal
Netherlands
Netherlands Antilles
New Caledonia
New Zealand
Nicaragua
Niger
Nigeria
Niue
Norfolk Island
Northern Mariana Islands
Norway
Oman
Pakistan
Palau
Panama
Papua New Guinea
Paraguay
Peru
Philippines
Pitcairn
Poland
Portugal

Qatar
Reunion
Romania
Russian Federation
Rwanda
S. Georgia and S. Sandwich Isls.
Saint Kitts and Nevis
Saint Lucia
Saint Vincent and The Grenadines
Samoa
San Marino
Sao Tome and Principe
Saudi Arabia
Senegal
Seychelles
Sierra Leone
Singapore
Slovak Republic
Slovenia
Solomon Islands
Somalia
South Africa
Spain
Sri Lanka
St. Helena
St. Pierre and Miquelon
Sudan
Suriname
Svalbard and Jan Mayen Islands
Swaziland
Sweden
Switzerland
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Togo
Tokelau
Tonga
Trinidad and Tobago
Tunisia
Turkey
Turkmenistan
Turks and Caicos Islands
Tuvalu
US Minor Outlying Islands
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States
Uruguay
Uzbekistan
Vanuatu
Vatican City State (Holy See)
Venezuela
Viet Nam
Virgin Islands (British)
Wallis and Futuna Islands
Western Sahara
Yemen
Yugoslavia
Zaire
Zambia
Zimbabwe

HTTP Status and Error Codes

Status codes provided by the REST API will have some minor variation these are defined here.

HTTP Status Code	Error Code	Condition
200	N/A	**Success** – Operation was completed successfully for the following operations: • GET • PUT
201	N/A	**Success** – Resource was created successfully for the following operation: • POST
400	10003	**Error** – Request could not be understood by the server.
	10006	**Error** – Operation requires an object_id.
	10301	**Error** – Illegal identifier specified.
	10302	**Error** – Invalid argument.
	11400	**Schema Error** – The parameters provided are incorrect.
	11401	**Schema Error** – Content-Type Not Supported.
	11402	**Schema Error** – Version is invalid.
401	11000	**Authentication Error** – Auth token is required.
	11001	**Authentication Error** – Invalid auth token.
	11002	**Authentication Error** – Auth token has expired.
403	11011	**Authorization Error** – Permission denied.
404	10000	**Error** – Service not found
	10300	**Error** – Object not found
405	10001	**Error** – Method not allowed
408	10002	**Error** – Request timed out
429	10005	**Error** – Too many requests
500	10100	**Error** – Internal Server Error
501	10200	**Error** – Not implemented
503	10004	**Error** – Service is temporarily unavailable

Terms of Service