virustotal3

Release 1.0.6

Jan 26, 2023

Contents:

CHAPTER 1

virustotal3 core

VirusTotal API v3 Core

Module to interact with the Core part of the API.

class core.**Domains**(*api_key=None*, *proxies=None*)

Class for the Domains endpoints

add_vote(*domain*, *verdict*, *timeout=None*)

Adds a verdict (vote) to a domain. The verdict can be either ‘malicious’ or ‘harmless’.

Parameters

- **domain** (*str*) – Domain
- **verdict** (*str*) – ‘malicious’ (-1) or ‘harmless’ (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the submitted vote.

get_relationship(*domain*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve objects related to a domain

Parameters

- **url** (*str*) – URL identifier
- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: communicating_files, downloaded_files, graphs, referrer_files, resolutions, siblings, subdomains, urls

  For further details, see: https://developers.virustotal.com/v3.0/reference#domains-relationships
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor

- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes (*domain*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve votes for a domain

Parameters

- **domain** (*str*) – Domain
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_domain (*domain*, *timeout=None*)
Retrieve information about a domain

Parameters

- **domain** (*str*) – Domain to scan
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the scan results.

class core.**Files** (*api_key=None*, *proxies=None*)
Class for the Files endpoints

add_comment (*file_hash*, *comment*, *timeout=None*)
Add a comment to a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **data** (*dict*) – Comment to add as dictionary. The package will take care of creating the JSON object.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the added comment.

add_vote (*file_hash*, *verdict*, *timeout=None*)
Adds a verdict (vote) to a file. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the submitted vote.

analyse_file (*file_hash*, *timeout=None*)
Re-analyse a file already in VirusTotal.

Parameters

- **file_hash** (*str*) – File hash to re-analyse
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the analysis ID.

download (*file_hash*, *output_dir='./'*, *timeout=None*)

Download a file for a given file hash.

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **output_dir** (*str, optional*) – Output directory, current working directory by default.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

get_comments (*file_hash*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve comments for a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the comments retrieved.

get_relationship (*file_id*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve an object related to a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **relationsip** (*str*) – Relationship object to retrieve. Can be one of the following:

  analyses, behaviours, bundled_files, carbonblack_children, carbonblack_parents, comments, compressed_parents, comments, contacted_domains, contacted_ips, contacted_urls, email_parents, embedded_domains, embedded_ips, execution_parents, graphs, itw_urls, overlay_parents, pcap_parents, pe_resource_parents, similar_files, submissions, screenshots, votes

  > For further details, see: https://developers.virustotal.com/v3.0/reference#files-relationships

- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the relationship object.

get_votes (*file_hash*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve votes for a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_file (*file_hash*, *timeout=None*)
Retrieve information on a file

Parameters

- **file_hash** (*str*) – File hash of the file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing information about the file.

upload (*sample*, *timeout=None*)
Upload a file.

The size of the file will be calculated and the endpoint to use will be determined based on the file size.

Parameters

- **sample** (*str*) – Path to file sample to upload
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the analysis ID

class core.**IP** (*api_key=None*, *proxies=None*)
Class for the IP Addresses endpoints

add_vote (*ip*, *verdict*, *timeout=None*)
Adds a verdict (vote) to a file. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **ip** (*str*) – IPv4 address
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the submitted vote.

get_relationship (*ip*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve information on a user for a given ip identifier.

Parameters

- **ip** (*str*) – IPv4 address

- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: communicating_files, downloaded_files, graphs, referrer_files, resolutions, siblings, subips, urls

  For further details, see: https://developers.virustotal.com/v3.0/reference#ips-relationships
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes (*ip*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve votes for a given IP address

Parameters

- **ip** (*str*) – IPv4 address
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the votes. The votes are located in the 'value' key.

info_ip (*ip*, *timeout=None*)

Retrieve information for a given IP address, such as AS owner, country, reputation, etc.

Parameters

- **ip** (*str*) – IPv4 address
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the scan results.

class core.**URL** (*api_key=None*, *proxies=None*)

Class for the URL endpoints

add_vote (*url*, *verdict*, *timeout=None*)

Add a verdict to a URL

Adds a verdict (vote) to a URL. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **url** (*str*) – URL identifier
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the submitted vote.

get_network_location (*url*, *timeout=None*)

Retrieve associated IPs and DNS records, site categories, and WHOIS info for a given URL.

Parameters

- **url** (*str*) – URL identifier

- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details of a URL, including its latest DNS records and IP addresses.

get_relationship (*url*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve information on an object for a given URL identifier.

Parameters

- **url** (*str*) – URL identifier
- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: analyses, downloaded_files, graphs, last_serving_ip_address, redirecting_urls, submissions
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes (*url*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve votes for a URL

Parameters

- **url** (*str*) – URL identifier
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_url (*url*, *timeout=None*)
Retrieve information about a URL. If the URL was previously scanned, results will be returned immediately. Otherwise, a URL scan will begin and results might take a few seconds to return.

Parameters

- **url** (*str*) – URL to scan
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the scan results.

core.**get_analysis** (*api_key*, *analysis_id*, *proxies=None*, *timeout=None*)
Retrieve information about an analysis

Parameters

- **api_key** (*str*) – VirusTotal API key
- **analysis_id** (*str*) – Analysis ID to retrieve
- **proxies** (*dict, optional*) – Dictionary containing proxies
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

1.1 get_analysis

`core.get_analysis`(*api_key*, *analysis_id*, *proxies=None*, *timeout=None*)

Retrieve information about an analysis

Parameters

- **api_key** (*str*) – VirusTotal API key
- **analysis_id** (*str*) – Analysis ID to retrieve
- **proxies** (*dict, optional*) – Dictionary containing proxies
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

1.2 Files

class `core.Files`(*api_key=None*, *proxies=None*)

Class for the Files endpoints

add_comment(*file_hash*, *comment*, *timeout=None*)

Add a comment to a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **data** (*dict*) – Comment to add as dictionary. The package will take care of creating the JSON object.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the added comment.

add_vote(*file_hash*, *verdict*, *timeout=None*)

Adds a verdict (vote) to a file. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the submitted vote.

analyse_file(*file_hash*, *timeout=None*)

Re-analyse a file already in VirusTotal.

Parameters

- **file_hash** (*str*) – File hash to re-analyse
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the analysis ID.

download (*file_hash*, *output_dir='./'*, *timeout=None*)
Download a file for a given file hash.

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **output_dir** (*str, optional*) – Output directory, current working directory by default.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

get_comments (*file_hash*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve comments for a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the comments retrieved.

get_relationship (*file_id*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve an object related to a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **relationsip** (*str*) – Relationship object to retrieve. Can be one of the following:

  analyses, behaviours, bundled_files, carbonblack_children, carbonblack_parents, comments, compressed_parents, comments, contacted_domains, contacted_ips, contacted_urls, email_parents, embedded_domains, embedded_ips, execution_parents, graphs, itw_urls, overlay_parents, pcap_parents, pe_resource_parents, similar_files, submissions, screenshots, votes

  > For further details, see: https://developers.virustotal.com/v3.0/reference#files-relationships

- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the relationship object.

get_votes (*file_hash*, *limit=None*, *cursor=None*, *timeout=None*)
Retrieve votes for a file

Parameters

- **file_hash** (*str*) – File hash (SHA256, MD5, SHA1)
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve

- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_file (*file_hash*, *timeout=None*)

Retrieve information on a file

Parameters

- **file_hash** (*str*) – File hash of the file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing information about the file.

upload (*sample*, *timeout=None*)

Upload a file.

The size of the file will be calculated and the endpoint to use will be determined based on the file size.

Parameters

- **sample** (*str*) – Path to file sample to upload
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the analysis ID

1.3 URL

class core.**URL** (*api_key=None*, *proxies=None*)

Class for the URL endpoints

add_vote (*url*, *verdict*, *timeout=None*)

Add a verdict to a URL

Adds a verdict (vote) to a URL. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **url** (*str*) – URL identifier
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the submitted vote.

get_network_location (*url*, *timeout=None*)

Retrieve associated IPs and DNS records, site categories, and WHOIS info for a given URL.

Parameters

- **url** (*str*) – URL identifier
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details of a URL, including its latest DNS records and IP addresses.

get_relationship(*url*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve information on an object for a given URL identifier.

Parameters

- **url** (*str*) – URL identifier
- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: analyses, downloaded_files, graphs, last_serving_ip_address, redirecting_urls, submissions
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes(*url*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve votes for a URL

Parameters

- **url** (*str*) – URL identifier
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_url(*url*, *timeout=None*)

Retrieve information about a URL. If the URL was previously scanned, results will be returned immediately. Otherwise, a URL scan will begin and results might take a few seconds to return.

Parameters

- **url** (*str*) – URL to scan
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the scan results.

1.4 Domains

class core.**Domains**(*api_key=None*, *proxies=None*)

Class for the Domains endpoints

add_vote(*domain*, *verdict*, *timeout=None*)

Adds a verdict (vote) to a domain. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **domain** (*str*) – Domain
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)

- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the submitted vote.

get_relationship (*domain, relationship, limit=None, cursor=None, timeout=None*)
Retrieve objects related to a domain

Parameters

- **url** (*str*) – URL identifier
- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: communicating_files, downloaded_files, graphs, referrer_files, resolutions, siblings, subdomains, urls

  For further details, see: https://developers.virustotal.com/v3.0/reference#domains-relationships
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes (*domain, limit=None, cursor=None, timeout=None*)
Retrieve votes for a domain

Parameters

- **domain** (*str*) – Domain
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the votes. The votes are located in the 'value' key.

info_domain (*domain, timeout=None*)
Retrieve information about a domain

Parameters

- **domain** (*str*) – Domain to scan
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the scan results.

1.5 IP

class core.**IP** (*api_key=None, proxies=None*)
Class for the IP Addresses endpoints

add_vote (*ip, verdict, timeout=None*)
Adds a verdict (vote) to a file. The verdict can be either 'malicious' or 'harmless'.

Parameters

- **ip** (*str*) – IPv4 address
- **verdict** (*str*) – 'malicious' (-1) or 'harmless' (+1)
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the submitted vote.

get_relationship(*ip*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve information on a user for a given ip identifier.

Parameters

- **relationship** (*str*) – Relationship object to retrieve. Can be one of the following: communicating_files, downloaded_files, graphs, referrer_files, resolutions, siblings, subips, urls

  For further details, see: https://developers.virustotal.com/v3.0/reference#ips-relationships
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

get_votes(*ip*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve votes for a given IP address

Parameters

- **ip** (*str*) – IPv4 address
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the votes. The votes are located in the 'value' key.

info_ip(*ip*, *timeout=None*)

Retrieve information for a given IP address, such as AS owner, country, reputation, etc.

Parameters

- **ip** (*str*) – IPv4 address
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict containing the scan results.

CHAPTER 2

virustotal3 enterprise

VirusTotal API v3 Enterprise

Module to interact with the Enterprise part of the API.

class enterprise.**Accounts**(*api_key=None*, *proxies=None*)

VT Enterprise Users & Groups

Manage and retrieve information on users and groups.

This part of the API still is under development by VirusTotal.

get_relationship(*group_id*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve information on a user for a given group ID. Currently, the only relationship object supported by the VirusTotal v3 API is *graphs*.

Parameters

- **group_id** (*str*) – User ID
- **relationship** (*str*) – Relationship
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

info_group(*group_id*, *timeout=None*)

Retrieve information on a group for a given ID

Parameters

- **group_id** (*str*) – User ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details on the group.

info_user (*user_id*, *timeout=None*)
Retrieve information on a user for a given ID

Parameters

- **user_id** (*str*) – User ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details on the user.

class enterprise.**Livehunt** (*api_key=None*, *proxies=None*)
VT Enterprise Livehunt Endpoints

Livehunt endpoints allowing to manage YARA rules and notifications.

api_key
VirusTotal API key

Type str

proxies
Dictionary with proxies

Type dict, optional

create_rulset (*data*, *timeout=None*)
Create a Livehunt ruleset

Parameters

- **data** (*dict*) – Rule to create.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the created rule.

delete_notification (*notification_id*, *timeout=None*)
Delete a notification for a given notification ID

Parameters

- **notification_id** (*str*) – Notification ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

delete_notifications (*tag*, *timeout=None*)
Delete notifications for a given tag

Parameters

- **tag** (*str*) – Notification tag
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

delete_ruleset (*ruleset_id*, *timeout=None*)
Delete ruleset

Delete ruleset for a given ID

Parameters

- **ruleset_id** (*str*) – Ruleset ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

get_notification_files (*limit=None*, *cursor=None*, *timeout=None*)
Retrieve file details and context attributes from notifications.

Parameters

- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple notifications.

get_notifications (*notification_id=None*, *limit=None*, *fltr=None*, *cursor=None*, *timeout=None*)
Retrieve a single notification for a given ID or all notifications at once.

Parameters

- **notification_id** (*str, optional*) – Notification ID required to return a single specific ruleset.
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **fltr** (*str, optional*) – Return the rulesets matching the given criteria only
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple notifications in JSON format.

get_rulesets (*ruleset_id=None*, *limit=None*, *fltr=None*, *order=None*, *cursor=None*, *timeout=None*)
Retrieve one or multiple rulesets

Retrieve a single ruleset for a given ID or all rulesets at once.

Parameters

- **ruleset_id** (*str, optional*) – Ruleset ID required to return a single specific ruleset
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **fltr** (*str, optional*) – Return the rulesets matching the given criteria only
- **order** (*str, optional*) – Sort order
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple rulesets.

update_ruleset (*ruleset_id*, *data*, *timeout=None*)
Update existing ruleset

Update an existing ruleset for a given ID

Parameters

- **ruleset_id** (*str*) – Ruleset ID
- **data** (*dict*) – Ruleset to update as dictionary. The package will take care of creating the JSON object.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the updated rule.

class enterprise.**Retrohunt**(*api_key=None*, *proxies=None*)

VirusTotal Retrohunt class

Run Retrohunting jobs.

abort_job(*job_id*, *timeout=None*)

Abort a job for a given ID

Parameters

- **job_id** (*str*) – Job ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

create_job(*data*, *timeout=None*)

Create a new Retrohunt job

Parameters **data** (*dict*) – Rule to create. See example below.

Returns A dict with the created rule.

delete_job(*job_id*, *timeout=None*)

Delete a job for a given ID

Parameters **job_id** (*str*) – Job ID

Returns None

get_jobs(*job_id=None*, *limit=None*, *fltr=None*, *cursor=None*, *timeout=None*)

Retrieve an existing Retrohunt jobs. Returns all jobs if no ID is specified.

Parameters

- **job_id** (*str, optional*) – Job ID
- **limit** (*int, optional*) – Maximum number of jobs to retrieve
- **fltr** (*str, optional*) – Filter matching specific jobs only
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one of multiple jobs.

get_matching_files(*job_id*, *timeout=None*)

Get matching files for a job ID

Parameters

- **job_id** (*str*) – Job ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with matching files

class enterprise.**ZipFiles**(*api_key=None*, *proxies=None*)

Zipping files

Zip and download an individual file or multiple files. Zip files are password protected.

This part of the API still is under development by VirusTotal.

create_zip(*data*, *timeout=None*)

Creates a password-protected ZIP file with files from VirusTotal.

Parameters

- **data** (*str*) – Dictionary with a list of hashes to download. See example request for dictionary: https://developers.virustotal.com/v3.0/reference#zip_files
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the progression and status of the archive compression process, including its ID. Use the info_zip() function to check the status of a Zip file for a given ID.

get_url(*zip_id*, *timeout=None*)

Get the download URL of a Zip file for a given ID. Will raise an exception if the file is not yet ready to download. Should be called only after info_zip() returns a 'finished' status.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns URL of the zip file to download

get_zip(*zip_id*, *output_dir*, *timeout=None*)

Download a zip file for a given ID.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **output_dir** (*str*) – Output directory where the file will be downloaded.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

info_zip(*zip_id*, *timeout=None*)

Check the status of a Zip file for a given ID.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the status of the zip file creation. When the value of the 'status' key is set to 'finished', the file is ready for download. Other status are: 'starting', 'creating', 'timeout', 'error-starting', 'error-creating'.

`enterprise.`**`file_feed`**`(`*api_key*, *time*, *timeout=None*`)`

Get a file feed batch for a given date, by the minute.

From the official documentation: "Time 201912010802 will return the batch corresponding to December 1st, 2019 08:02 UTC. You can download batches up to 7 days old, and the most recent batch has always a 60 minutes lag with respect with to the current time."

Parameters

- **`api_key`** (*`str`*) – VirusTotal key
- **`time`** (*`str`*) – YYYYMMDDhhmm
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns each line is a json string for one report

Return type StringIO

`enterprise.`**`search`**`(`*api_key*, *query*, *order=None*, *limit=None*, *cursor=None*, *descriptors_only=None*, *proxies=None*, *timeout=None*`)`

Search for files and return the file details.

Parameters

- **`api_key`** (*`str`*) – VirusTotal API key
- **`query`** (*`str`*) – Search query
- **`order`** (*`str, optional`*) – Sort order. Can be one of the following: size, positives, last_submission_date, first_submission_date, times_submitted. Can be followed by a + or -. Default is 'last_submission_date-'
- **`limit`** (*`int, optional`*) – Maximum number of results to retrieve
- **`cursor`** (*`str, optional`*) – Continuation cursor
- **`descriptors_only`** (*`bool, optional`*) – Return file descriptor only instead of all details
- **`proxies`** (*`dict, optional`*) – Dictionary with proxies
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the results from the search.

`enterprise.`**`url_feed`**`(`*api_key*, *time*, *timeout=None*`)`

Get a URL feed batch for a given date, by the minute.

From the official documentation: "Time 201912010802 will return the batch corresponding to December 1st, 2019 08:02 UTC. You can download batches up to 7 days old, and the most recent batch has always a 60 minutes lag with respect with to the current time."

Parameters

- **`api_key`** (*`str`*) – VirusTotal key
- **`time`** (*`str`*) – YYYYMMDDhhmm
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns each line is a json string for one report

Return type StringIO

2.1 search

`enterprise.search(api_key, query, order=None, limit=None, cursor=None, descriptors_only=None, proxies=None, timeout=None)`

Search for files and return the file details.

Parameters

- **api_key** (*str*) – VirusTotal API key
- **query** (*str*) – Search query
- **order** (*str, optional*) – Sort order. Can be one of the following: size, positives, last_submission_date, first_submission_date, times_submitted. Can be followed by a + or -. Default is 'last_submission_date-'
- **limit** (*int, optional*) – Maximum number of results to retrieve
- **cursor** (*str, optional*) – Continuation cursor
- **descriptors_only** (*bool, optional*) – Return file descriptor only instead of all details
- **proxies** (*dict, optional*) – Dictionary with proxies
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the results from the search.

2.2 file_feed

`enterprise.file_feed(api_key, time, timeout=None)`

Get a file feed batch for a given date, by the minute.

From the official documentation: "Time 201912010802 will return the batch corresponding to December 1st, 2019 08:02 UTC. You can download batches up to 7 days old, and the most recent batch has always a 60 minutes lag with respect with to the current time."

Parameters

- **api_key** (*str*) – VirusTotal key
- **time** (*str*) – YYYYMMDDhhmm
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns each line is a json string for one report

Return type StringIO

2.3 Livehunt

`class enterprise.Livehunt(api_key=None, proxies=None)`

VT Enterprise Livehunt Endpoints

Livehunt endpoints allowing to manage YARA rules and notifications.

api_key
VirusTotal API key

Type str

proxies
Dictionary with proxies

Type dict, optional

create_rulset (*data*, *timeout=None*)
Create a Livehunt ruleset

Parameters

- **data** (*dict*) – Rule to create.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the created rule.

delete_notification (*notification_id*, *timeout=None*)
Delete a notification for a given notification ID

Parameters

- **notification_id** (*str*) – Notification ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

delete_notifications (*tag*, *timeout=None*)
Delete notifications for a given tag

Parameters

- **tag** (*str*) – Notification tag
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

delete_ruleset (*ruleset_id*, *timeout=None*)
Delete ruleset

Delete ruleset for a given ID

Parameters

- **ruleset_id** (*str*) – Ruleset ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns None

get_notification_files (*limit=None*, *cursor=None*, *timeout=None*)
Retrieve file details and context attributes from notifications.

Parameters

- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **cursor** (*str, optional*) – Continuation cursor

- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple notifications.

get_notifications (*notification_id=None*, *limit=None*, *fltr=None*, *cursor=None*, *timeout=None*)
Retrieve a single notification for a given ID or all notifications at once.

Parameters

- **notification_id** (*str, optional*) – Notification ID required to return a single specific ruleset.
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **fltr** (*str, optional*) – Return the rulesets matching the given criteria only
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple notifications in JSON format.

get_rulesets (*ruleset_id=None*, *limit=None*, *fltr=None*, *order=None*, *cursor=None*, *timeout=None*)
Retrieve one or multiple rulesets

Retrieve a single ruleset for a given ID or all rulesets at once.

Parameters

- **ruleset_id** (*str, optional*) – Ruleset ID required to return a single specific ruleset
- **limit** (*int, optional*) – Maximum number of rulesets to retrieve
- **fltr** (*str, optional*) – Return the rulesets matching the given criteria only
- **order** (*str, optional*) – Sort order
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one or multiple rulesets.

update_ruleset (*ruleset_id*, *data*, *timeout=None*)
Update existing ruleset

Update an existing ruleset for a given ID

Parameters

- **ruleset_id** (*str*) – Ruleset ID
- **data** (*dict*) – Ruleset to update as dictionary. The package will take care of creating the JSON object.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the updated rule.

2.4 Retrohunt

class `enterprise.`**`Retrohunt`**(*api_key=None*, *proxies=None*)

VirusTotal Retrohunt class

Run Retrohunting jobs.

`abort_job`(*job_id*, *timeout=None*)

Abort a job for a given ID

Parameters

- **`job_id`** (*`str`*) – Job ID
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns None

`create_job`(*data*, *timeout=None*)

Create a new Retrohunt job

Parameters **`data`** (*`dict`*) – Rule to create. See example below.

Returns A dict with the created rule.

`delete_job`(*job_id*, *timeout=None*)

Delete a job for a given ID

Parameters **`job_id`** (*`str`*) – Job ID

Returns None

`get_jobs`(*job_id=None*, *limit=None*, *fltr=None*, *cursor=None*, *timeout=None*)

Retrieve an existing Retrohunt jobs. Returns all jobs if no ID is specified.

Parameters

- **`job_id`** (*`str, optional`*) – Job ID
- **`limit`** (*`int, optional`*) – Maximum number of jobs to retrieve
- **`fltr`** (*`str, optional`*) – Filter matching specific jobs only
- **`cursor`** (*`str, optional`*) – Continuation cursor
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with one of multiple jobs.

`get_matching_files`(*job_id*, *timeout=None*)

Get matching files for a job ID

Parameters

- **`job_id`** (*`str`*) – Job ID
- **`timeout`** (*`float, optional`*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with matching files

2.5 Accounts

class enterprise.**Accounts**(*api_key=None*, *proxies=None*)

VT Enterprise Users & Groups

Manage and retrieve information on users and groups.

This part of the API still is under development by VirusTotal.

get_relationship(*group_id*, *relationship*, *limit=None*, *cursor=None*, *timeout=None*)

Retrieve information on a user for a given group ID. Currently, the only relationship object supported by the VirusTotal v3 API is *graphs*.

Parameters

- **group_id** (*str*) – User ID
- **relationship** (*str*) – Relationship
- **limit** (*str, optional*) – Limit of results to return
- **cursor** (*str, optional*) – Continuation cursor
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the relationship object.

info_group(*group_id*, *timeout=None*)

Retrieve information on a group for a given ID

Parameters

- **group_id** (*str*) – User ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details on the group.

info_user(*user_id*, *timeout=None*)

Retrieve information on a user for a given ID

Parameters

- **user_id** (*str*) – User ID
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the details on the user.

2.6 ZipFiles

class enterprise.**ZipFiles**(*api_key=None*, *proxies=None*)

Zipping files

Zip and download an individual file or multiple files. Zip files are password protected.

This part of the API still is under development by VirusTotal.

create_zip(*data*, *timeout=None*)

Creates a password-protected ZIP file with files from VirusTotal.

Parameters

- **data** (*str*) – Dictionary with a list of hashes to download. See example request for dictionary: https://developers.virustotal.com/v3.0/reference#zip_files
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the progression and status of the archive compression process, including its ID. Use the info_zip() function to check the status of a Zip file for a given ID.

get_url(*zip_id*, *timeout=None*)

Get the download URL of a Zip file for a given ID. Will raise an exception if the file is not yet ready to download. Should be called only after info_zip() returns a 'finished' status.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns URL of the zip file to download

get_zip(*zip_id*, *output_dir*, *timeout=None*)

Download a zip file for a given ID.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **output_dir** (*str*) – Output directory where the file will be downloaded.
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

info_zip(*zip_id*, *timeout=None*)

Check the status of a Zip file for a given ID.

Parameters

- **zip_id** (*str*) – ID of the zip file
- **timeout** (*float, optional*) – The amount of time in seconds the request should wait before timing out.

Returns A dict with the status of the zip file creation. When the value of the 'status' key is set to 'finished', the file is ready for download. Other status are: 'starting', 'creating', 'timeout', 'error-starting', 'error-creating'.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index