OpenCTI client for Python

Release 4.2.0

OpenCTI Project

May 19, 2023

CONTENTS:

The pycti library is designed to help OpenCTI users and developers to interact with the OpenCTI platform GraphQL API.

The Python library requires Python >= 3.

CHAPTER

ONE

GETTING STARTED

1.1 Installation

Please install the latest pycti version available from PyPI:

```
$ pip3 install pycti
```

1.2 Using the helper functions

The main class `OpenCTIApiClient` contains all what you need to interact with the platform, you just have to initialize it. If your OpenCTI instance requires mTLS, you can specify the paths to your cert and private key in a tuple with the *cert* parameter when initializing the *OpenCTIApiClient*. If you need to specify a path to a CA root certificate, you can do so with the *ssl_verify* parameter.

The following example shows how you create an indicator in OpenCTI using the python library with TLP marking and OpenCTI compatible date format.

```
from dateutil.parser import parse
from pycti import OpenCTIApiClient
from stix2 import TLP_GREEN

# OpenCTI API client initialization
opencti_api_client = OpenCTIApiClient("https://myopencti.server", "mysupersecrettoken
↪")

# Define an OpenCTI compatible date
date = parse("2019-12-01").strftime("%Y-%m-%dT%H:%M:%SZ")

# Get the OpenCTI marking for stix2 TLP_GREEN
TLP_GREEN_CTI = opencti_api_client.marking_definition.read(id=TLP_GREEN["id"])

# Use the client to create an indicator in OpenCTI
indicator = opencti_api_client.indicator.create(
    name="C2 server of the new campaign",
    description="This is the C2 server of the campaign",
    pattern_type="stix",
    pattern="[domain-name:value = 'www.5z8.info']",
    x_opencti_main_observable_type="IPv4-Addr",
    valid_from=date,
    update=True,
    markingDefinitions=[TLP_GREEN_CTI["id"]],
)
```

CHAPTER

TWO

PYCTI

2.1 Functions

- *get_config_variable()*: [summary]

pycti.**get_config_variable**(*env_var*, *yaml_path*, *config={}*, *isNumber=False*, *default=None*)
[summary]

Parameters

- **env_var** (str) – environment variable name
- **yaml_path** (List) – path to yaml config
- **config** (Dict) – client config dict, defaults to {}
- **isNumber** (Optional[bool]) – specify if the variable is a number, defaults to False

Return type Union[bool, int, None, str]

2.2 Classes

- *AttackPattern*: Undocumented.
- *Campaign*: Undocumented.
- *CaseIncident*: Undocumented.
- *CaseRfi*: Undocumented.
- *CaseRft*: Undocumented.
- *ConnectorType*: An enumeration.
- *CourseOfAction*: Undocumented.
- *DataComponent*: Undocumented.
- *DataSource*: Undocumented.
- *ExternalReference*: Undocumented.

- *Feedback*: Undocumented.
- *Grouping*: Undocumented.
- *Identity*: Undocumented.
- *Incident*: Undocumented.
- *Indicator*: Main Indicator class for OpenCTI
- *Infrastructure*: Main Infrastructure class for OpenCTI
- *IntrusionSet*: Undocumented.
- *KillChainPhase*: Undocumented.
- *Label*: Undocumented.
- *Location*: Undocumented.
- *Malware*: Undocumented.
- *MarkingDefinition*: Undocumented.
- *Note*: Undocumented.
- *ObservedData*: Undocumented.
- *OpenCTIApiClient*: Main API client for OpenCTI
- *OpenCTIApiConnector*: OpenCTIApiConnector
- *OpenCTIApiWork*: OpenCTIApiJob
- *OpenCTIConnector*: Main class for OpenCTI connector
- *OpenCTIConnectorHelper*: Python API for OpenCTI connector
- *OpenCTIStix2*: Python API for Stix2 in OpenCTI
- *OpenCTIStix2Splitter*: Undocumented.
- *OpenCTIStix2Update*: Python API for Stix2 Update in OpenCTI
- *OpenCTIStix2Utils*: Undocumented.
- *Opinion*: Undocumented.
- *Report*: Undocumented.
- *StixCoreRelationship*: Undocumented.
- *StixCyberObservable*: Undocumented.
- *StixNestedRefRelationship*: Undocumented.
- *StixCyberObservableTypes*: An enumeration.
- *StixDomainObject*: Undocumented.
- *StixMetaTypes*: An enumeration.
- *MultipleRefRelationship*: An enumeration.
- *StixObjectOrStixRelationship*: Undocumented.
- *StixSightingRelationship*: Undocumented.
- *ThreatActor*: Main ThreatActor class for OpenCTI
- *Tool*: Undocumented.

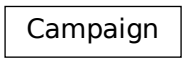
- *Vulnerability*: Undocumented.

class pycti.**AttackPattern**(*opencti*)

Inheritance

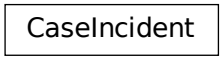
class pycti.**Campaign**(*opencti*)

Inheritance



class pycti.**CaseIncident**(*opencti*)

Inheritance

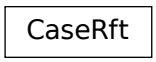


class pycti.**CaseRfi**(*opencti*)

Inheritance

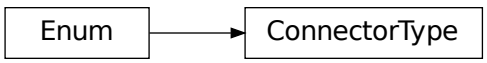
class pycti.**CaseRft**(*opencti*)

Inheritance



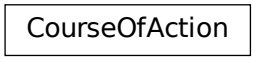
class pycti.**ConnectorType**(*value*)
An enumeration.

Inheritance



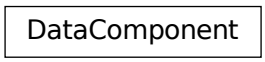
class pycti.**CourseOfAction**(*opencti*)

Inheritance



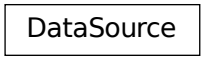
class pycti.**DataComponent**(*opencti*)

Inheritance



class pycti.**DataSource**(*opencti*)

Inheritance

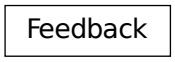


class pycti.**ExternalReference**(*opencti*, *file*)

Inheritance

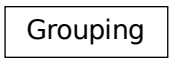
class pycti.**Feedback**(*opencti*)

Inheritance



class pycti.**Grouping**(*opencti*)

Inheritance

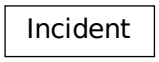


class pycti.**Identity**(*opencti*)

Inheritance

class pycti.**Incident**(*opencti*)

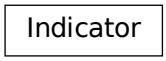
Inheritance



class pycti.**Indicator**(*opencti*)
Main Indicator class for OpenCTI

Parameters **opencti** – instance of `OpenCTIApiClient`

Inheritance



add_stix_cyber_observable(***kwargs*)
Add a Stix-Cyber-Observable object to Indicator object (based-on)

Parameters

- **id** – the id of the Indicator
- **indicator** – Indicator object
- **stix_cyber_observable_id** – the id of the Stix-Observable

Returns Boolean True if there has been no import error

create(***kwargs*)
Create an Indicator object

Parameters

- **name** (*str*) – the name of the Indicator
- **pattern** (*str*) – stix indicator pattern
- **x_opencti_main_observable_type** (*str*) – type of the observable

Returns Indicator object

Return type *Indicator*

import_from_stix2(***kwargs*)
Import an Indicator object from a STIX2 object

Parameters

- **stixObject** – the Stix-Object Indicator
- **extras** – extra dict
- **update** (*bool*) – set the update flag on import

Returns Indicator object

Return type *Indicator*

list(***kwargs*)
List Indicator objects

The list method accepts the following kwargs:

Parameters

- **filters** (*list*) – (optional) the filters to apply
- **search** (*str*) – (optional) a search keyword to apply for the listing
- **first** (*int*) – (optional) return the first n rows from the *after* ID or the beginning if not set
- **after** (*str*) – (optional) OpenCTI object ID of the first row for pagination
- **orderBy** (*str*) – (optional) the field to order the response on
- **orderMode** (*bool*) – (optional) either "*asc*" or "*desc*"
- **customAttributes** (*list*) – (optional) list of attributes keys to return
- **getAll** (*bool*) – (optional) switch to return all entries (be careful to use this without any other filters)
- **withPagination** (*bool*) – (optional) switch to use pagination

Returns List of Indicators

Return type list

read(***kwargs*)

Read an Indicator object

read can be either used with a known OpenCTI entity *id* or by using a valid filter to search and return a single Indicator entity or None.

The list method accepts the following kwargs.

Note: either *id* or *filters* is required.

Parameters

- **id** (*str*) – the id of the Threat-Actor
- **filters** (*list*) – the filters to apply if no id provided

Returns Indicator object

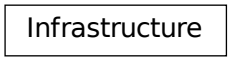
Return type *Indicator*

class pycti.**Infrastructure**(*opencti*)

Main Infrastructure class for OpenCTI

Parameters **opencti** – instance of `OpenCTIApiClient`

Inheritance



list(***kwargs*)

List Infrastructure objects

The list method accepts the following kwargs:

Parameters

- **filters** (*list*) – (optional) the filters to apply
- **search** (*str*) – (optional) a search keyword to apply for the listing
- **first** (*int*) – (optional) return the first n rows from the *after* ID or the beginning if not set
- **after** (*str*) – (optional) OpenCTI object ID of the first row for pagination
- **orderBy** (*str*) – (optional) the field to order the response on
- **orderMode** (*bool*) – (optional) either "*asc*" or "*desc*"
- **customAttributes** (*list*) – (optional) list of attributes keys to return
- **getAll** (*bool*) – (optional) switch to return all entries (be careful to use this without any other filters)
- **withPagination** (*bool*) – (optional) switch to use pagination

read(***kwargs*)

Read an Infrastructure object

read can be either used with a known OpenCTI entity *id* or by using a valid filter to search and return a single Infrastructure entity or None.

The list method accepts the following kwargs.

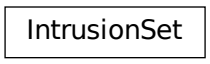
Note: either *id* or *filters* is required.

Parameters

- **id** (*str*) – the id of the Threat-Actor
- **filters** (*list*) – the filters to apply if no id provided

class pycti.**IntrusionSet**(*opencti*)

Inheritance

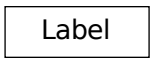


class pycti.**KillChainPhase**(*opencti*)

Inheritance

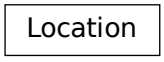
class pycti.**Label**(*opencti*)

Inheritance



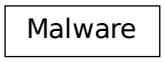
class pycti.**Location**(*opencti*)

Inheritance



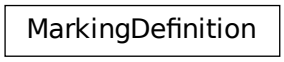
class pycti.**Malware**(*opencti*)

Inheritance



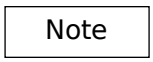
class pycti.**MarkingDefinition**(*opencti*)

Inheritance



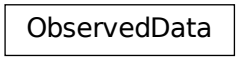
class pycti.**Note**(*opencti*)

Inheritance



class pycti.**ObservedData**(*opencti*)

Inheritance



class pycti.**OpenCTIApiClient**(*url*, *token*, *log_level='info'*, *ssl_verify=False*, *proxies=None*, *json_logging=False*, *cert=None*)

Main API client for OpenCTI

Parameters

- **url** (*str*) – OpenCTI API url
- **token** (*str*) – OpenCTI API token
- **log_level** (*str, optional*) – log level for the client
- **ssl_verify** (*bool, optional*) –
- **proxies** –
- **json_logging** (*bool, optional*) – format the logs as json if set to True

Inheritance

fetch_opencti_file(*fetch_uri*, *binary=False*, *serialize=False*)
get file from the OpenCTI API

Parameters

- **fetch_uri** (*str*) – download URI to use
- **binary** (*bool, optional*) – [description], defaults to False

Returns returns either the file content as text or bytes based on *binary*

Return type str or bytes

get_logs_worker_config()
get the logsWorkerConfig

return: the logsWorkerConfig rtype: dict

get_stix_content(*id*)
get the STIX content of any entity

return: the STIX content in JSON rtype: dict

health_check()
submit an example request to the OpenCTI API.

Returns returns *True* if the health check has been successful

Return type bool

log(*level*, *message*)
log a message with defined log level :param level: must be a valid logging log level (debug, info, warning, error) :type level: str :param message: the message to log :type message: str

not_empty(*value*)
check if a value is empty for str, list and int

Parameters **value** (*str or list or int or float or bool or datetime.date*) – value to check

Returns returns *True* if the value is one of the supported types and not empty

Return type bool

process_multiple(*data*, *with_pagination=False*)
processes data returned by the OpenCTI API with multiple entities

Parameters

- **data** (dict) – data to process
- **with_pagination** – whether to use pagination with the API

Return type Union[dict, list]

Returns returns either a dict or list with the processes entities

process_multiple_fields(*data*)
processes data returned by the OpenCTI API with multiple fields

Parameters **data** (*dict*) – data to process

Returns returns the data dict with all fields processed

Return type dict

process_multiple_ids(*data*)
processes data returned by the OpenCTI API with multiple ids

Parameters **data** – data to process

Return type list

Returns returns a list of ids

query(*query*, *variables={}*)
submit a query to the OpenCTI GraphQL API

Parameters

- **query** (*str*) – GraphQL query string
- **variables** (*dict, optional*) – GraphQL query variables, defaults to {}

Returns returns the response json content

Return type Any

upload_file(***kwargs*)
upload a file to OpenCTI API

Parameters ****kwargs** – arguments for file upload (required: *file_name* and *data*)

Returns returns the query respons for the file upload

Return type dict

upload_pending_file(***kwargs*)
upload a file to OpenCTI API

Parameters ****kwargs** – arguments for file upload (required: *file_name* and *data*)

Returns returns the query response for the file upload

Return type dict

class pycti.**OpenCTIApiConnector**(*api*)

Inheritance

list()
list available connectors

Returns return dict with connectors

Return type dict

ping(*connector_id*, *connector_state*)
pings a connector by id and state

Parameters

- **connector_id** (*str*) – the connectors id
- **connector_state** (Any) – state for the connector

Returns the response pingConnector data dict

Return type dict

register(*connector*)
register a connector with OpenCTI

Parameters **connector** (OpenCTIConnector) – *OpenCTIConnector* connector object

Returns the response registerConnector data dict

Return type dict

unregister(*_id*)
unregister a connector with OpenCTI

Parameters **_id** (*string*) – *OpenCTIConnector* connector id

Returns the response registerConnector data dict

Return type dict

class pycti.**OpenCTIApiWork**(*api*)
OpenCTIApiJob

Inheritance

class pycti.**OpenCTIConnector**(*connector_id*, *connector_name*, *connector_type*, *scope*, *auto*, *only_contextual*)
Main class for OpenCTI connector

Parameters

- **connector_id** (*str*) – id for the connector (valid uuid4)
- **connector_name** (*str*) – name for the connector
- **connector_type** (*str*) – valid OpenCTI connector type (see *ConnectorType*)

- **scope** (*str*) – connector scope

Raises **ValueError** – if the connector type is not valid

Inheritance

to_input()
connector input to use in API query

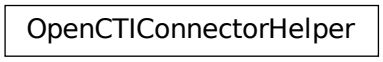
Returns dict with connector data

Return type dict

class pycti.**OpenCTIConnectorHelper**(*config*)
Python API for OpenCTI connector

Parameters **config** (*Dict*) – dict standard config

Inheritance



static check_max_tlp(*tlp*, *max_tlp*)
check the allowed TLP levels for a TLP string

Parameters

- **tlp** (*str*) – string for TLP level to check
- **max_tlp** (*str*) – the highest allowed TLP level

Returns TLP level in allowed TLPs

Return type bool

date_now()
get the current date (UTC) :return: current datetime for utc :rtype: str

date_now_z()
get the current date (UTC) :return: current datetime for utc :rtype: str

get_state()
get the connector state

Returns returns the current state of the connector if there is any

Return type

listen(*message_callback*)
listen for messages and register callback function

Parameters **message_callback** (*Callable[[Dict], str]*) – callback function to process messages

Return type None

listen_stream(*message_callback*, *url=None*, *token=None*, *verify_ssl=None*, *start_timestamp=None*, *live_stream_id=None*, *listen_delete=None*, *no_dependencies=None*, *recover_iso_date=None*, *with_inferences=None*)
listen for messages and register callback function

Parameters **message_callback** – callback function to process messages

Return type ListenStream

send_stix2_bundle(*bundle*, ***kwargs*)
send a stix2 bundle to the API

Parameters

- **work_id** – a valid work id
- **bundle** – valid stix2 bundle
- **entities_types** (*list, optional*) – list of entities, defaults to None
- **update** (*bool, optional*) – whether to updated data in the database, defaults to False

Raises **ValueError** – if the bundle is empty

Returns list of bundles

Return type list

set_state(*state*)
sets the connector state

Parameters **state** (*Dict or None*) – state object

Return type None

static stix2_create_bundle(*items*)
create a stix2 bundle with items

Parameters **items** – valid stix2 items

Returns JSON of the stix2 bundle

Return type

static stix2_deduplicate_objects(*items*)
deduplicate stix2 items

Parameters **items** – valid stix2 items

Returns de-duplicated list of items

Return type list

stix2_get_embedded_objects(*item*)
gets created and marking refs for a stix2 item

Parameters **item** – valid stix2 item

Returns returns a dict of created_by of object_marking_refs

Return type Dict

stix2_get_entity_objects(*entity*)
process a stix2 entity

Parameters **entity** – valid stix2 entity

Returns entity objects as list

Return type list

stix2_get_relationship_objects(*relationship*)
get a list of relations for a stix2 relationship object

Parameters **relationship** – valid stix2 relationship

Returns list of relations objects

Return type list

stix2_get_report_objects(*report*)
get a list of items for a stix2 report object

Parameters **report** – valid stix2 report object

Returns list of items for a stix2 report object

Return type list

class pycti.**OpenCTIStix2**(*opencti*)
Python API for Stix2 in OpenCTI

Parameters **opencti** – OpenCTI instance

Inheritance

check_max_marking_definition(*max_marking_definition_entity*, *entity_marking_definitions*)
checks if a list of marking definitions conforms with a given max level

Parameters

- **max_marking_definition_entity**(*str, optional*) – the maximum allowed marking definition level
- **entity_marking_definitions** (*list*) – list of entities to check

Returns *True* if the list conforms with max marking definition

Return type bool

convert_markdown(*text*)
converts input text to markdown style code annotation

Parameters **text** (*str*) – input text

Returns sanitized text with markdown style code annotation

Return type str

extract_embedded_relationships(*stix_object*, *types=None*)
extracts embedded relationship objects from a stix2 entity

Parameters

- **stix_object** (Dict) – valid stix2 object
- **types** (*list, optional*) – list of stix2 types, defaults to None

Returns embedded relationships as dict

Return type dict

filter_objects(*uuids*, *objects*)
filters objects based on UUIDs

Parameters

- **uuids** (*list*) – list of UUIDs
- **objects** (*list*) – list of objects to filter

Returns list of filtered objects

Return type list

format_date(*date=None*)
converts multiple input date formats to OpenCTI style dates

Parameters **date** (*Any [datetime, date, str or none]*) – input date

Returns OpenCTI style date

Return type string

import_bundle_from_file(*file_path*, *update=False*, *types=None*)
import a stix2 bundle from a file

Parameters

- **file_path** (*str*) – valid path to the file
- **update** (*bool, optional*) – whether to updated data in the database, defaults to False
- **types** (*list, optional*) – list of stix2 types, defaults to None

Returns list of imported stix2 objects

Return type List

import_bundle_from_json(*json_data*, *update=False*, *types=None*, *retry_number=None*)
import a stix2 bundle from JSON data

Parameters

- **json_data** (Union[str, bytes]) – JSON data
- **update** (*bool, optional*) – whether to updated data in the database, defaults to False

- **types** (*list, optional*) – list of stix2 types, defaults to None

Returns list of imported stix2 objects

Return type List

import_object(*stix_object*, *update=False*, *types=None*)
import a stix2 object

Parameters

- **stix_object** (Dict) – valid stix2 object
- **update** (*bool, optional*) – whether to updated data in the database, defaults to False
- **types** (*list, optional*) – list of stix2 types, defaults to None

Returns list of imported stix2 objects

Return type list

pick_aliases(*stix_object*)
check stix2 object for multiple aliases and return a list

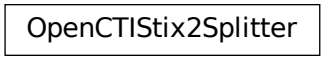
Parameters **stix_object** (Dict) – valid stix2 object

Returns list of aliases

Return type list

class pycti.**OpenCTIStix2Splitter**

Inheritance



split_bundle(*bundle*, *use_json=True*, *event_version=None*)
splits a valid stix2 bundle into a list of bundles :param bundle: valid stix2 bundle :type bundle: :param use_json: is JSON? :type use_json: :raises Exception: if data is not valid JSON :return: returns a list of bundles :rtype: list

static stix2_create_bundle(*bundle_id*, *bundle_seq*, *items*, *use_json*, *event_version=None*)
create a stix2 bundle with items

Parameters

- **items** – valid stix2 items
- **use_json** – use JSON?

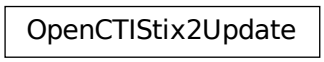
Returns JSON of the stix2 bundle

Return type

class pycti.**OpenCTIStix2Update**(*opencti*)
Python API for Stix2 Update in OpenCTI

Parameters **opencti** – OpenCTI instance

Inheritance

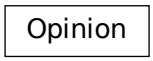


class pycti.**OpenCTIStix2Utils**

Inheritance

class pycti.**Opinion**(*opencti*)

Inheritance



class pycti.**Report**(*opencti*)

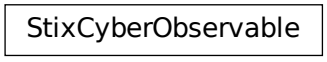
Inheritance

class pycti.**StixCoreRelationship**(*opencti*)

Inheritance

class pycti.**StixCyberObservable**(*opencti*, *file*)

Inheritance

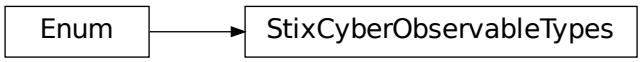


class pycti.**StixNestedRefRelationship**(*opencti*)

Inheritance

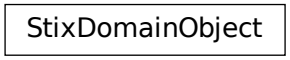
class pycti.**StixCyberObservableTypes**(*value*)
An enumeration.

Inheritance



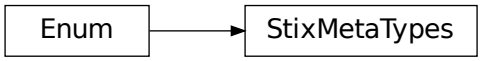
class pycti.**StixDomainObject**(*opencti*, *file*)

Inheritance



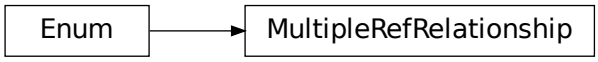
class pycti.**StixMetaTypes**(*value*)
An enumeration.

Inheritance



class pycti.**MultipleRefRelationship**(*value*)
An enumeration.

Inheritance

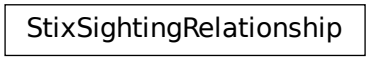


class pycti.**StixObjectOrStixRelationship**(*opencti*)

Inheritance

class pycti.**StixSightingRelationship**(*opencti*)

Inheritance



class pycti.**ThreatActor**(*opencti*)
Main ThreatActor class for OpenCTI

Parameters **opencti** – instance of `OpenCTIApiClient`

Inheritance

create(***kwargs*)

Create a Threat-Actor object

The Threat-Actor entity will only be created if it doesn't exists By setting *update* to *True* it acts like an upsert and updates fields of an existing Threat-Actor entity.

The create method accepts the following kwargs.

Note: *name* and *description* or *stix_id* is required.

Parameters

- **stix_id** (*str*) – stix2 id reference for the Threat-Actor entity
- **createdBy** (*str*) – (optional) id of the organization that created the knowledge
- **objectMarking** (*list*) – (optional) list of OpenCTI markin definition ids
- **objectLabel** (*list*) – (optional) list of OpenCTI label ids
- **externalReferences** (*list*) – (optional) list of OpenCTI external references ids
- **revoked** (*bool*) – is this entity revoked
- **confidence** (*int*) – confidence level
- **lang** (*str*) – language
- **created** (*str*) – (optional) date in OpenCTI date format
- **modified** (*str*) – (optional) date in OpenCTI date format
- **name** (*str*) – name of the threat actor
- **description** (*str*) – description of the threat actor
- **aliases** (*list*) – (optional) list of alias names for the Threat-Actor
- **threat_actor_types** (*list*) – (optional) list of threat actor types
- **first_seen** (*str*) – (optional) date in OpenCTI date format
- **last_seen** (*str*) – (optional) date in OpenCTI date format
- **roles** (*list*) – (optional) list of roles
- **goals** (*list*) – (optional) list of goals
- **sophistication** (*str*) – (optional) describe the actors sophistication in text
- **resource_level** (*str*) – (optional) describe the actors resource_level in text
- **primary_motivation** (*str*) – (optional) describe the actors primary_motivation in text

- **secondary_motivations** (*list*) – (optional) describe the actors secondary_motivations in list of string
- **personal_motivations** (*list*) – (optional) describe the actors personal_motivations in list of strings
- **update** (*bool*) – (optional) choose to updated an existing Threat-Actor entity, default *False*

list(***kwargs*)

List Threat-Actor objects

The list method accepts the following kwargs:

Parameters

- **filters** (*list*) – (optional) the filters to apply
- **search** (*str*) – (optional) a search keyword to apply for the listing
- **first** (*int*) – (optional) return the first n rows from the *after* ID or the beginning if not set
- **after** (*str*) – (optional) OpenCTI object ID of the first row for pagination
- **orderBy** (*str*) – (optional) the field to order the response on
- **orderMode** (*bool*) – (optional) either "*asc*" or "*desc*"
- **getAll** (*bool*) – (optional) switch to return all entries (be careful to use this without any other filters)
- **withPagination** (*bool*) – (optional) switch to use pagination

Return type dict

read(***kwargs*)

Read a Threat-Actor object

read can be either used with a known OpenCTI entity *id* or by using a valid filter to search and return a single Threat-Actor entity or None.

The list method accepts the following kwargs.

Note: either *id* or *filters* is required.

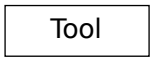
Parameters

- **id** (*str*) – the id of the Threat-Actor
- **filters** (*list*) – the filters to apply if no id provided

Return type Optional[dict]

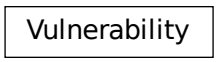
class pycti.**Tool**(*opencti*)

Inheritance



class pycti.**Vulnerability**(*opencti*)

Inheritance



CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

M

N

O

P

Q

R

S

T

U

V