

---

# OpenCTI client for Python

*Release 4.2.0*

**OpenCTI Project**

**Jun 06, 2022**



**CONTENTS:**

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Using the helper functions . . . . .	3
<b>2</b>	<b>pycti</b>	<b>5</b>
2.1	Functions . . . . .	5
2.2	Classes . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>
	<b>Index</b>	<b>33</b>



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

The Python library requires Python  $\geq 3$ .



## 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. 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",
    indicator_pattern="[domain-name:value = 'www.5z8.info']",
    x_opencti_main_observable_type="IPv4-Addr",
    valid_from=date,
    update=True,
    markingDefinitions=[TLP_GREEN_CTI["id"]],
)
```





- *Functions*
- *Classes*

## 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) – environnement 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.
- *ConnectorType*: An enumeration.
- *CourseOfAction*: Undocumented.
- *ExternalReference*: 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.
- *StixCyberObservableRelationship*: Undocumented.
- *StixCyberObservableTypes*: An enumeration.
- *StixDomainObject*: Undocumented.
- *StixMetaTypes*: An enumeration.
- *MultipleStixCyberObservableRelationship*: An enumeration.
- *StixObjectOrStixRelationship*: Undocumented.
- *StixSightingRelationship*: Undocumented.
- *ThreatActor*: Main ThreatActor class for OpenCTI
- *Tool*: Undocumented.
- *Vulnerability*: Undocumented.

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

### Inheritance

AttackPattern

```
class pycti.Campaign(opencti)
```

### Inheritance

Campaign

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

### Inheritance



```
class pycti.CourseOfAction(opencti)
```

### Inheritance

CourseOfAction

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

#### Inheritance

ExternalReference

```
class pycti.Identity(opencti)
```

#### Inheritance

Identity

```
class pycti.Incident(opencti)
```

#### Inheritance

Incident

```
class pycti.Indicator(opencti)
```

Main Indicator class for OpenCTI

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

## Inheritance

Indicator

**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

Infrastructure

**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` (*openti*)

#### Inheritance

IntrusionSet

**class** `pycti.KillChainPhase` (*openti*)

#### Inheritance

KillChainPhase

```
class pycti.Label(opencti)
```

#### Inheritance

Label

```
class pycti.Location(opencti)
```

#### Inheritance

Location

```
class pycti.Malware(opencti)
```

#### Inheritance

Malware

```
class pycti.MarkingDefinition(opencti)
```



## Inheritance

MarkingDefinition

```
class pycti.Note(opencti)
```

## Inheritance

Note

```
class pycti.ObservedData(opencti)
```

## Inheritance

ObservedData

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

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

OpenCTIApiClient

**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

**Parameters**

- **level** (*str*) – must be a valid logging log level (debug, info, warning, error)
- **message** (*str*) – the message to log

**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 respons for the file upload

**Return type** dict

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

## Inheritance

OpenCTIApiConnector

**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

OpenCTIApiWork

```
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

OpenCTIConnector

```
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

OpenCTIConnectorHelper

**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=True*,  
*no\_dependencies=False*, *recover\_iso\_date=None*, *with\_inferences=False*)  
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*) – 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` (*opentcti*)

Python API for Stix2 in OpenCTI

**Parameters** `opentcti` – OpenCTI instance

## Inheritance

OpenCTIStix2

**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

OpenCTIStix2Splitter

**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

OpenCTIStix2Update

**class** `pycti.OpenCTIStix2Utils`

### Inheritance

OpenCTIStix2Utils

```
class pycti.Opinion(opencti)
```

### Inheritance

Opinion

```
class pycti.Report(opencti)
```

### Inheritance

Report

```
class pycti.StixCoreRelationship(opencti)
```

### Inheritance

StixCoreRelationship

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

### Inheritance

StixCyberObservable

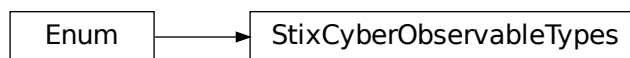
```
class pycti.StixCyberObservableRelationship(openciti)
```

### Inheritance

StixCyberObservableRelationship

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

### Inheritance



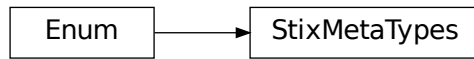
```
class pycti.StixDomainObject(openciti, file)
```

### Inheritance

StixDomainObject

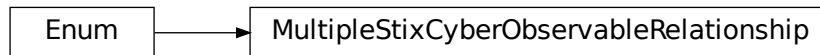
```
class pycti.StixMetaTypes(value)  
    An enumeration.
```

### Inheritance



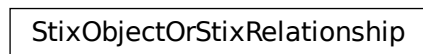
```
class pycti.MultipleStixCyberObservableRelationship(value)  
    An enumeration.
```

### Inheritance



```
class pycti.StixObjectOrStixRelationship(opencti)
```

### Inheritance



```
class pycti.StixSightingRelationship(opencti)
```

## Inheritance

StixSightingRelationship

```
class pycti.ThreatActor(opencti)
```

Main ThreatActor class for OpenCTI

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

## Inheritance

ThreatActor

```
create(**kwargs)
```

Create a Threat-Actor object

The Threat-Actor entity will only be created if it doesn't exist. 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 marking 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



```
graph TD; Tool[Tool];
```

```
class pycti.Vulnerability(opencti)
```

### Inheritance



```
graph TD; Vulnerability[Vulnerability];
```



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### p

pycti, 5



## A

`add_stix_cyber_observable()` (*pycti.Indicator method*), 9  
*AttackPattern* (class in *pycti*), 6

## C

*Campaign* (class in *pycti*), 7  
`check_max_marking_definition()` (*pycti.OpenCTISix2 method*), 20  
`check_max_tlp()` (*pycti.OpenCTIConnectorHelper static method*), 18  
*ConnectorType* (class in *pycti*), 7  
`convert_markdown()` (*pycti.OpenCTISix2 method*), 20  
*CourseOfAction* (class in *pycti*), 7  
`create()` (*pycti.Indicator method*), 9  
`create()` (*pycti.ThreatActor method*), 26

## D

`date_now()` (*pycti.OpenCTIConnectorHelper method*), 18  
`date_now_z()` (*pycti.OpenCTIConnectorHelper method*), 18

## E

*ExternalReference* (class in *pycti*), 7  
`extract_embedded_relationships()` (*pycti.OpenCTISix2 method*), 20

## F

`fetch_opencti_file()` (*pycti.OpenCTIApiClient method*), 14  
`filter_objects()` (*pycti.OpenCTISix2 method*), 20  
`format_date()` (*pycti.OpenCTISix2 method*), 20

## G

`get_config_variable()` (in module *pycti*), 5  
`get_logs_worker_config()` (*pycti.OpenCTIApiClient method*), 14  
`get_state()` (*pycti.OpenCTIConnectorHelper method*), 18

`get_stix_content()` (*pycti.OpenCTIApiClient method*), 14

## H

`health_check()` (*pycti.OpenCTIApiClient method*), 14

## I

*Identity* (class in *pycti*), 8  
`import_bundle_from_file()` (*pycti.OpenCTISix2 method*), 21  
`import_bundle_from_json()` (*pycti.OpenCTISix2 method*), 21  
`import_from_stix2()` (*pycti.Indicator method*), 9  
`import_object()` (*pycti.OpenCTISix2 method*), 21  
*Incident* (class in *pycti*), 8  
*Indicator* (class in *pycti*), 8  
*Infrastructure* (class in *pycti*), 10  
*IntrusionSet* (class in *pycti*), 11

## K

*KillChainPhase* (class in *pycti*), 11

## L

*Label* (class in *pycti*), 11  
`list()` (*pycti.Indicator method*), 9  
`list()` (*pycti.Infrastructure method*), 10  
`list()` (*pycti.OpenCTIApiConnector method*), 16  
`list()` (*pycti.ThreatActor method*), 27  
`listen()` (*pycti.OpenCTIConnectorHelper method*), 18  
`listen_stream()` (*pycti.OpenCTIConnectorHelper method*), 18  
*Location* (class in *pycti*), 12  
`log()` (*pycti.OpenCTIApiClient method*), 14

## M

*Malware* (class in *pycti*), 12  
*MarkingDefinition* (class in *pycti*), 12  
module  
*pycti*, 5

MultipleStixCyberObservableRelationship  
(class in pycti), 25

## N

not\_empty() (pycti.OpenCTIApiClient method), 14  
Note (class in pycti), 13

## O

ObservedData (class in pycti), 13  
OpenCTIApiClient (class in pycti), 13  
OpenCTIApiConnector (class in pycti), 15  
OpenCTIApiWork (class in pycti), 16  
OpenCTIConnector (class in pycti), 17  
OpenCTIConnectorHelper (class in pycti), 17  
OpenCTISTix2 (class in pycti), 19  
OpenCTISTix2Splitter (class in pycti), 21  
OpenCTISTix2Update (class in pycti), 22  
OpenCTISTix2Utils (class in pycti), 22  
Opinion (class in pycti), 23

## P

pick\_aliases() (pycti.OpenCTISTix2 method), 21  
ping() (pycti.OpenCTIApiConnector method), 16  
process\_multiple() (pycti.OpenCTIApiClient  
method), 14  
process\_multiple\_fields()  
(pycti.OpenCTIApiClient method), 15  
process\_multiple\_ids()  
(pycti.OpenCTIApiClient method), 15  
pycti  
module, 5

## Q

query() (pycti.OpenCTIApiClient method), 15

## R

read() (pycti.Indicator method), 10  
read() (pycti.Infrastructure method), 11  
read() (pycti.ThreatActor method), 27  
register() (pycti.OpenCTIApiConnector method),  
16  
Report (class in pycti), 23

## S

send\_stix2\_bundle()  
(pycti.OpenCTIConnectorHelper method),  
18  
set\_state() (pycti.OpenCTIConnectorHelper  
method), 19  
split\_bundle() (pycti.OpenCTISTix2Splitter  
method), 22  
stix2\_create\_bundle()  
(pycti.OpenCTIConnectorHelper static  
method), 19

stix2\_create\_bundle()  
(pycti.OpenCTISTix2Splitter static method), 22  
stix2\_deduplicate\_objects()  
(pycti.OpenCTIConnectorHelper static  
method), 19  
stix2\_get\_embedded\_objects()  
(pycti.OpenCTIConnectorHelper method),  
19  
stix2\_get\_entity\_objects()  
(pycti.OpenCTIConnectorHelper method),  
19  
stix2\_get\_relationship\_objects()  
(pycti.OpenCTIConnectorHelper method),  
19  
stix2\_get\_report\_objects()  
(pycti.OpenCTIConnectorHelper method),  
19  
StixCoreRelationship (class in pycti), 23  
StixCyberObservable (class in pycti), 23  
StixCyberObservableRelationship (class in  
pycti), 24  
StixCyberObservableTypes (class in pycti), 24  
StixDomainObject (class in pycti), 24  
StixMetaTypes (class in pycti), 24  
StixObjectOrStixRelationship (class in  
pycti), 25  
StixSightingRelationship (class in pycti), 25

## T

ThreatActor (class in pycti), 26  
to\_input() (pycti.OpenCTIConnector method), 17  
Tool (class in pycti), 28

## U

unregister() (pycti.OpenCTIApiConnector  
method), 16  
upload\_file() (pycti.OpenCTIApiClient method),  
15  
upload\_pending\_file() (pycti.OpenCTIApiClient  
method), 15

## V

Vulnerability (class in pycti), 28