CSIRTG SDK Documentation

Release 0.0

CSIRT Gadgets

December 26, 2016

Contents

Contents:

CHAPTER 1

client

class csirtgsdk.client.**Client**(*remote='https://csirtg.io/api'*, *token=None*, *proxy=None*, *timeout=300*, *verify_ssl=True*)

Bases: object

get(*uri*, *params={}*)

HTTP GET function

Parameters

- **uri** – REST endpoint
- **params** – optional HTTP params to pass to the endpoint

Returns list of results (usually a list of dicts)

Example: ret = cli.get('/search', params={ 'q': 'example.org' })

post(*uri*, *data*)

HTTP POST function

Parameters

- **uri** – REST endpoint to POST to
- **data** – list of dicts to be passed to the endpoint

Returns list of dicts, usually will be a list of objects or id's

Example: ret = cli.post('/indicators', { 'indicator': 'example.com' })

submit_bulk(*indicators*, *user*, *feed*)

Submit action against the IndicatorBulk endpoint

Parameters

- **indicators** – list of Indicator Objects
- **user** – feed username
- **feed** – feed name

Returns list of Indicator Objects submitted

from csirtgsdk.client import Client from csirtgsdk.indicator import Indicator

remote = 'https://csirtg.io/api' token = '' verify_ssl = True

i = { 'indicator': 'example.com', 'feed': 'test', 'user': 'admin', 'comment': 'this is a test',

```
}
data = []
cli = Client(remote=remote, token=token, verify_ssl=verify_ssl)
for x in range(0, 5):
    data.append(  Indicator(cli, i)
    )
ret = cli.submit_bulk(data, 'csirtgadgets', 'test-feed')
```

CHAPTER 2

search

class csirtgsdk.search.**Search**(*client*)

Bases: object

Search Object class

search(*q*, *limit=None*)

Performs a search against the /search endpoint

Parameters

- **q** – query to be searched for [STRING]
- **limit** – limit the results [INT]

Returns list of dicts

CHAPTER 3

feed

class csirtgsdk.feed.**Feed**(*client*)

Bases: object

Represents a Feed Object

index(*user*)

Returns a list of Feeds from the API

Parameters **user** – feed username

Returns list

Example: ret = feed.index('csirtgadgets')

new(*user*, *name*, *description=None*)

Creates a new Feed object

Parameters

- **user** – feed username
- **name** – feed name
- **description** – feed description

Returns dict

remove(*user*, *name*)

Removes a feed

Parameters

- **user** – feed username
- **name** – feed name

Returns true/false

show(*user*, *name*, *limit=None*, *lasttime=None*)

Returns a specific Feed from the API

Parameters

- **user** – feed username
- **name** – feed name
- **limit** – limit the results

- **lasttime** – only show >= lasttime

Returns dict

Example: ret = feed.show('csirtgadgets', 'port-scanners', limit=5)

CHAPTER 4

indicator

class indicator.**Indicator**(*client*, *args*)

Bases: object

Represents an Indicator object

comments(*user*, *feed*, *id*)

Return comments for a specific indicator id

Parameters

- **user** – feed username
- **feed** – feed name
- **id** – indicator id [INT]

Returns list

Example: ret = Indicator.comments('csirtgadgets','port-scanners', '1234')

show(*user*, *feed*, *id*)

Show a specific indicator by id

Parameters

- **user** – feed username
- **feed** – feed name
- **id** – indicator endpoint id [INT]

Returns dict

Example: ret = Indicator.show('csirtgadgets','port-scanners', '1234')

submit()

Submit action on the Indicator object

Returns Indicator Object

CHAPTER 5

utils

class utils.**Map**(**args*, ***kwargs*)

Bases: dict

Example: m = Map({'first_name': 'Eduardo'}, last_name='Pool', age=24, sports=['Soccer'])

Reference: http://stackoverflow.com/questions/2352181/how-to-use-a-dot-to-access-members-of-dictionary

utils.**read_config**(*args*)

Reads in an ArgParse objet with args.confg as the YAML style config path

Parameters **args** – ArgParse object

Returns dict of options based on ArgParse and the YAML config

utils.**setup_logging**(*args*)

Sets up basic logging

Parameters **args** – ArgParse arguments

Returns nothing. sets logger up globally

CHAPTER 6

Examples

```
$ csirtg --search example.com
$ csirtg --user csirtgadgets --feeds
$ csirtg --user csirtgadgets --feed uce-urls
$ csirtg --user csirtgadgets --new --feed scanners --description 'a feed of port scanners'
$ csirtg --user csirtgadgets --feed scanners --new --indicator 1.1.1.1 --tags scanner --comment 'this
```

CHAPTER 7

Search

```
from csirtgsdk.client import Client
from csirtgsdk.search import Search
from pprint import pprint

# Initiate client object
cli = Client(token=token)

# Search for an indicator
ret = Search(cli).search('example.org', limit=5)

# short form
ret = Search(Client(token=token)).search('example.org', limit=5)

# pretty print the returned data structure
pprint(ret)
```

CHAPTER 8

Show Feed

```
from csirtgsdk.client import Client
from csirtgsdk.feed import Feed
from pprint import pprint

# Initiate client object
cli = Client(token=token)

# Pull a feed
ret = Feed(cli).show('csirtgadgets', 'uce-urls')

# pprint the returned data structure
pprint(ret)
```

CHAPTER 9

Create Feed

```
from csirtgsdk.client import Client
from csirtgsdk.feed import Feed
from pprint import pprint

# Initiate client object
cli = Client(token=token)

# Create a feed
ret = Feed(cli).new('csirtgadgets', 'scanners', description='a feed of port scanners')

# pprint the returned data structure
pprint(ret)
```

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

u

C

F

G

I

M

N

P

R

S

U