
Censys Python

Release 1.1.1

Censys Team

Feb 23, 2021

TABLE OF CONTENTS

1 Quick Start	3
2 Usage	5
3 Advanced Usage	9
4 ASM Usage	11
5 CLI Reference	17
6 API Reference	19
7 Development	43
8 Testing	45
9 Contributing	47
Python Module Index	49
Index	51

An easy-to-use and lightweight API wrapper for the Censys Search Engine (censys.io).

QUICK START

Assuming you have Python already, install the package:

from PyPi

from GitHub

from Kali Linux

```
$ pip install censys
```

```
$ pip install git+https://github.com/censys/censys-python@main
```

```
$ sudo apt install python3-censys
```

Configure your credentials:

Search API

ASM API

```
$ censys config
```

```
$ censys config-asm
```


USAGE

The Censys Search API provides functionality for interacting with Censys resources such as IPv4 addresses, Websites, and Certificates, and for viewing Account information such as query quota.

There are six API options that this library provides access to:

- `search` - Allows searches against the IPv4 addresses, Websites, and Certificates indexes using the same search syntax as the [web app](#).
- `view` - Returns the structured data we have about a specific IPv4 address, Website, or Certificate, given the resource's natural ID.
- `report` - Allows you to view resources as a spectrum based on attributes of the resource, similar to the [Report Builder page](#) on the web app.
- `data` - Returns collections of scan series whose metadata includes a description of the data collected in the series and links to the individual scan results.
- `account` - Returns information about your Censys account, including your current query quota usage. This function is available for all index types.
- `bulk` - Returns the structured data for certificates in bulk, given the certificates' SHA-256 fingerprints.

More details about each option can be found in the Censys API documentation: <https://censys.io/api>. A list of index fields can be found in the Censys API definitions page: <https://censys.io/ipv4/help/definitions>.

Python class objects must be initialized for each resource index (IPv4 addresses, Websites, and Certificates).

- `CensysIPv4`
- `CensysWebsites`
- `CensysCertificates`

2.1 search

Below we show an example using the `CensysIPv4` index.

```
import censys.ipv4

c = censys.ipv4.CensysIPv4()

for page in c.search(
    "443.https.get.headers.server: Apache AND location.country: Japan",
    max_records=10
):
```

(continues on next page)

(continued from previous page)

```
print(page)

# You can optionally restrict the (resource-specific) fields to be
# returned in the matching results. Default behavior is to return a map
# including `location` and `protocol`.
fields = [
    "ip",
    "updated_at",
    "443.https.get.title",
    "443.https.get.headers.server",
    "443.https.get.headers.x_powered_by",
    "443.https.get.metadata.description",
    "443.https.tls.certificate.parsed.subject_dn",
    "443.https.tls.certificate.parsed.names",
    "443.https.tls.certificate.parsed.subject.common_name",
]

for page in c.search(
    "443.https.get.headers.server: Apache AND location.country: Japan",
    fields,
    max_records=10,
):
    print(page)
```

2.2 view

Below we show an example using the CensysCertificates index.

```
import censys.certificates

c = censys.certificates.CensysCertificates()

# View specific certificate
cert = c.view("a762bf68f167f6fbdf2ab00fdefeb8b96f91335ad6b483b482dfd42c179be076")
print(cert)
```

2.3 report

Below we show an example using the CensysWebsites index.

```
import censys.websites

c = censys.websites.CensysWebsites()

# The report method constructs a report using a query, an aggregation field, and the
# number of buckets to bin.
websites = c.report(
    """ "welcome to" AND tags.raw: "http" """,
    field="80.http.get.headers.server.raw",
    buckets=5,
)
print(websites)
```

2.4 data

Below we show an example using the CensysData index.

```
import censys.data

c = censys.data.CensysData()

# View a specific result from a specific series
result = c.view_result("ipv4_2018", "20200818")
print(result)
```

2.5 account

Below we show an example using the CensysIPv4 index.

```
import censys.ipv4

c = censys.ipv4.CensysIPv4()

# Gets account data
account = c.account()
print(account)

# Gets account quota
quota = c.quota()
print(quota)
```

2.6 bulk

Please note this method is only available only for the certificate index

Below we show an example using the CensysCertificates index.

```
import censys.certificates

c = censys.certificates.CensysCertificates()

fingerprints = [
    "fce621c0dc1c666d03d660472f636ce91e66e96460545f0da7eb1a24873e2f70",
    "a762bf68f167f6fbdf2ab00fdefeb8b96f91335ad6b483b482dfd42c179be076"
]

# Get bulk certificate data
certs = c.bulk(fingerprints)
print(certs)
```


ADVANCED USAGE

3.1 Proxies

If you need to use a proxy, you can configure resource indexes with the proxies argument:

```
import censys.ipv4

proxies = {
    "https": "http://10.10.1.10:1080",
}

c = censys.ipv4.CensysIPv4(proxies=proxies)

for page in c.search(
    "443.https.get.headers.server: Apache AND location.country: Japan", max_records=10
):
    print(page)
```

Note: HTTP proxies will be ignored in favor of HTTPS proxies.

See [Requests Proxies](#) for more information on the format of proxies.

ASM USAGE

The Censys ASM API provides functionality for interacting with Censys ASM endpoints such as Seeds, Assets, and Logbook Events.

There are three API options that this library provides access to:

- `seeds` - Provides programmatic management of seeds in the ASM platform.
- `assets` - Returns asset data for hosts, certificates, and domains. This option also allows the user to manage tags and comments on assets.
- `events` - Returns logbook events. Can be used to execute targeted searches for events based on start id or date, and event type filters.

More details about each option can be found in the Censys ASM API documentation: <https://app.censys.io/api-docs>. Users can also test example requests from the API documentation page.

Python class objects can be used individually, but must be initialized for each resource type (Seeds, Assets, and Events).

- `Seeds()`
- `Assets("ASSET_TYPE ["hosts" | "certificates" | "domains"]",)`
- `Events()`

Alternatively, all three class objects can be used together by initializing an `AsmClient` object. This client wraps the three APIs under one object for ease of use.

- `AsmClient()`

4.1 Seeds

Below we show examples for **listing seeds** from the Censys ASM platform.

```
from censys.asm.seeds import Seeds

s = Seeds()

# Get all seeds
seeds = s.get_seeds()
print(seeds)

# Get a specific type of seed. Optional seed types are ["IP_ADDRESS", "DOMAIN_NAME",
↪ "CIDR", "ASN"]
# Here we get IP address seeds.
```

(continues on next page)

(continued from previous page)

```
seeds = s.get_seeds('IP_ADDRESS')
print(seeds)

# Get a single seed by its ID (here we get seed with ID=3)
seeds = s.get_seeds(3)
print(seeds)
```

Below we show examples for **adding seeds** to the Censys ASM platform.

```
from censys.asm.seeds import Seeds

s = Seeds()

# Add a list of seeds. To add a single seed, just pass a list containing one seed.
# Here, we add two ASN seeds.
seed_list = [
    {"type": "ASN", "value": 99998, "label": "seed-test-label"},
    {"type": "ASN", "value": 99999, "label": "seed-test-label"}
]
s.add_seeds(seed_list)

# Add a list of seeds, replacing existing seeds with a specified label
# Here, all seeds with label="seed-test-label" will be removed and then
# Seeds 99996 and 99997 will be added.
seed_list = [
    {"type": "ASN", "value": 99996},
    {"type": "ASN", "value": 99997}
]
s.replace_seeds_by_label("seed-test-label", seed_list)
```

Below we show examples for **deleting seeds** from the Censys ASM platform.

```
from censys.asm.seeds import Seeds

s = Seeds()

# Delete all seeds with a specified label
# Here we delete all seeds with label="seed-test-label"
s.delete_seeds_by_label("seed-test-label")

# Delete a seed by its ID
# Here, a seed with ID=224 will be deleted.
s.delete_seed_by_id(224)
```

4.2 Assets

There are three types of assets (Hosts, Certificates, Domains). Each asset type shares the same API interface so we will use a mixture of asset types in the following examples.

Below we show examples for **viewing assets** on the Censys ASM platform.

```
from censys.asm.assets import Assets

h = Assets("hosts")
```

(continues on next page)

(continued from previous page)

```
# Get a generator that returns hosts
hosts = h.get_assets()
print(next(hosts))

# Get a single host by ID (here we get host with ID="0.0.0.0")
host = h.get_asset_by_id("0.0.0.0")
print(host)
```

Below we show examples for **managing asset comments** via the ASM API.

```
from censys.asm.assets import Assets

d = Assets("domains")

# Get a generator that returns all comments on a specific domain asset
# Here we get all comments on the domain with ID="my_domain.com"
comments = d.get_comments("my_domain.com")
print(next(comments))

# Get a single comment on a specific domain by comment ID
# Here we look at domain with ID="my_domain.com" and get comment with ID=3
comment = d.get_comment_by_id("my_domain.com", 3)
print(comment)

# Add a comment to a domain asset
# Here we add comment "hello world" to domain with ID="my_domain.com"
d.add_comment("my_domain.com", "hello world")
```

Below we show examples for **managing asset tags** via the ASM API.

```
from censys.asm.assets import Assets

c = Assets("certificates")
cert_sha = "0006afclddc8431aa57c812adf028ab4f168b25bf5f06e94af86edbafa88dfe0"

# Add a tag to a certificate asset
# Here we add tag "New" to certificate with ID=cert_sha
c.add_tag(cert_sha, "New")

# We can optionally give the tag a hexadecimal color where the default=#ffffff (white)
# Here we add a blue tag "New-2" to certificate with ID=cert_sha
c.add_tag(cert_sha, "New-2", color="#0011ff")

# Delete a tag by tag name
# Here we delete tag name="New" from certificate with ID=cert_sha
c.delete_tag(cert_sha, "New")
```

4.3 Events

Note: Note that all timestamp fields in logbook operations use **ISO-8601** format. This is the full list of event types that can be used as filters:

- CERT
- CERT_RISK
- DOMAIN
- DOMAIN_EXPIRATION_DATE
- DOMAIN_MAIL_EXCHANGE_SERVER
- DOMAIN_NAME_SERVER
- DOMAIN_REGISTRAR
- DOMAIN_RISK
- DOMAIN_SUBDOMAIN
- HOST
- HOST_CERT
- HOST_PORT
- HOST_PROTOCOL
- HOST_RISK
- HOST_SOFTWARE
- HOST_VULNERABILITY

Below we show examples for **creating a logbook cursor** for retrieving filtered events.

```
from censys.asm.events import Events

e = Events()

# Get a logbook cursor beginning at timestamp "2020-04-22T06:55:01.000Z"
cursor = e.get_cursor("2020-04-22T06:55:01.000Z")
print(cursor)

# Get a logbook cursor beginning at event ID=10
cursor = e.get_cursor(10)
print(cursor)

# Get a logbook cursor that filters on events of type "CERT" and "CERT_RISK"
cursor = e.get_cursor(filters=["CERT", "CERT_RISK"])
print(cursor)

# Get a logbook cursor combining previous start ID and filters
cursor = e.get_cursor(10, filters=["CERT", "CERT_RISK"])
print(cursor)
```

Below we show examples for **getting logbook events**.

```
from censys.asm.events import Events

e = Events()

# Get a generator that returns all events
events = e.get_events()
print(next(events))

# Get events based off cursor specifications
events = e.get_events(cursor)
print(next(events))
```

4.4 AsmClient

The Censys AsmClient wraps the Seeds, Assets, and Events classes into a single object. It can be used as a single point of interaction for all three APIs.

Below we show how to initialize the AsmClient class object as well as a couple examples of its use. Note that with the AsmClient object, all Seeds, Assets, and Event operations can be accessed in a similar way as the individual APIs above.

```
from censys.asm.client import AsmClient

client = AsmClient()

# Get all seeds
seeds = client.seeds.get_seeds()
print(seeds)

# Get all domain assets
domains = client.domains.get_assets()
print(next(domains))

# Get all events
events = client.events.get_events()
print(next(events))
```


CLI REFERENCE

5.1 censys

```
usage: censys [-h] {search,hnri,config,config-asm} ...
```

-h, --help
show this help message and exit

5.1.1 censys config

Configure Censys Search API Settings

```
usage: censys config [-h]
```

-h, --help
show this help message and exit

5.1.2 censys config-asm

Configure Censys ASM API Settings

```
usage: censys config-asm [-h]
```

-h, --help
show this help message and exit

5.1.3 censys hnri

Home Network Risk Identifier (H.N.R.I.)

```
usage: censys hnri [-h] [--api-id API_ID] [--api-secret API_SECRET]
```

-h, --help
show this help message and exit

--api-id <api_id>
a Censys API ID (alternatively you can use the env variable CENSYS_API_ID)

--api-secret <api_secret>
a Censys API SECRET (alternatively you can use the env variable CENSYS_API_SECRET)

5.1.4 censys search

Query Censys Search for resource data by providing a query string, the resource index, and the fields to be returned

```
usage: censys search [-h] [--api-id API_ID] [--api-secret API_SECRET] -q QUERY
                    [--index-type ipv4|certs|websites]
                    [--fields FIELDS [FIELDS ...]] [--overwrite]
                    [-f json|csv|screen] [-o OUTPUT]
                    [--start-page START_PAGE] [--max-pages MAX_PAGES]
```

-h, --help

show this help message and exit

--api-id <api_id>

a Censys API ID (alternatively you can use the env variable CENSYS_API_ID)

--api-secret <api_secret>

a Censys API SECRET (alternatively you can use the env variable CENSYS_API_SECRET)

-q <query>, **--query** <query>

a string written in Censys Search syntax

--index-type {ipv4,certs,websites}

which resource index to query

--fields <fields>

list of index-specific fields

--overwrite

overwrite instead of append fields returned by default with fields provided in the fields argument

-f <json|csv|screen>, **--format** <json|csv|screen>

format of output

-o <output>, **--output** <output>

output file path

--start-page <start_page>

page number to start from

--max-pages <max_pages>

maximum number of pages of results to return

API REFERENCE

6.1 Subpackages

6.1.1 censys.asm package

Submodules

censys.asm.api module

Base for interacting with the Censys ASM API.

class `censys.asm.api.CensysAsmAPI` (*api_key: Optional[str] = None, **kwargs*)
Bases: `censys.base.CensysAPIBase`

This is the base class for ASM's Seeds, Assets, and Events classes

DEFAULT_URL: `str = 'https://app.censys.io/api/v1'`
Default ASM API base URL.

censys.asm.assets module

Interact with the Censys Assets API.

class `censys.asm.assets.Assets` (*asset_type, api_key: Optional[str] = None, **kwargs*)
Bases: `censys.asm.api.CensysAsmAPI`

Assets API class

add_comment (*asset_id: str, comment: str*) → dict
Adds a comment to a specified asset on the ASM platform.

Parameters

- **asset_id** (*str*) – Asset ID to add comment to.
- **comment** (*str*) – New comment text.

Returns Added comment results.

Return type dict

add_tag (*asset_id: str, name: str, color: Optional[str] = None*) → dict
Adds a tag to a specified asset on the ASM platform.

Parameters

- **asset_id** (*str*) – Asset ID to add tag to.

- **name** (*str*) – New tag name.
- **color** (*str*, *optional*) – New tag color.

delete_tag (*asset_id: str*, *name: str*) → dict

Deletes a tag from a specified asset on the ASM platform by tag name.

Parameters

- **asset_id** (*str*) – Asset ID to delete tag from.
- **name** (*str*) – Tag name to delete.

get_asset_by_id (*asset_id: str*) → dict

Requests asset data by ID.

Parameters **asset_id** (*str*) – Requested asset ID.

Returns Asset search result.

Return type dict

get_assets (*page_number: int = 1*, *page_size: Optional[int] = None*) → Generator[dict, None, None]

Requests assets data.

Parameters

- **page_number** (*int*, *optional*) – Page number to begin at when searching.
- **page_size** (*int*, *optional*) – Page size for retrieving assets.

Returns Asset search results.

Return type generator

get_comment_by_id (*asset_id: str*, *comment_id: int*) → dict

Requests a comment on a specified asset by comment ID.

Parameters

- **asset_id** (*str*) – Asset ID for requested comments.
- **comment_id** (*int*) – Requested comment ID.

Returns Comment search result.

Return type dict

get_comments (*asset_id: str*, *page_number: int = 1*, *page_size: Optional[int] = None*) → Generator[dict, None, None]

Requests comments on a specified asset.

Parameters

- **asset_id** (*str*) – Asset ID for requested comments.
- **page_number** (*int*, *optional*) – Page number to begin at when searching.
- **page_size** (*int*, *optional*) – Page size for retrieving comments.

Returns Comment search results.

Return type generator

`censys.asm.assets.format_tag` (*name: str*, *color: Optional[str] = None*) → dict

Formats tag name and color request data.

Parameters

- **name** (*str*) – Tag name.

- **color** (*str, optional*) – Tag color.

Returns Formatted tag request data.

Return type dict

censys.asm.client module

Interact with the Censys Seeds, Assets, and Logbook APIs.

class `censys.asm.client.AsmClient` (*api_key: Optional[str] = None, **kwargs*)

Bases: object

Client ASM API class.

censys.asm.events module

Interact with the Censys Logbook API.

class `censys.asm.events.Events` (*api_key: Optional[str] = None, **kwargs*)

Bases: `censys.asm.api.CensysAsmAPI`

Events API class

get_cursor (*start: Optional[Union[datetime.datetime, int]] = None, filters: Optional[List[str]] = None*) → str

Requests a logbook cursor.

Parameters

- **start** (*datetime or int, optional*) – Timestamp or event ID to begin searching.
- **filters** (*list, optional*) – List of filters applied to logbook search results.

Returns Cursor result.

Return type str

get_events (*cursor: Optional[str] = None*) → Generator[dict, None, None]

Requests logbook events from inception or from the provided cursor.

Parameters **cursor** (*str, optional*) – Logbook cursor.

Returns Logbook events results.

Return type generator

class `censys.asm.events.Filters`

Bases: object

Logbook filters class

CERT = 'CERT'

CERT_RISK = 'CERT_RISK'

DOMAIN = 'DOMAIN'

DOMAIN_EXPIRATION_DATE = 'DOMAIN_EXPIRATION_DATE'

DOMAIN_MAIL_EXCHANGE_SERVER = 'DOMAIN_MAIL_EXCHANGE_SERVER'

DOMAIN_NAME_SERVER = 'DOMAIN_NAME_SERVER'

```
DOMAIN_REGISTRAR = 'DOMAIN_REGISTRAR'  
DOMAIN_RISK = 'DOMAIN_RISK'  
DOMAIN_SUBDOMAIN = 'DOMAIN_SUBDOMAIN'  
HOST = 'HOST'  
HOST_CERT = 'HOST_CERT'  
HOST_PORT = 'HOST_PORT'  
HOST_PROTOCOL = 'HOST_PROTOCOL'  
HOST_RISK = 'HOST_RISK'  
HOST_SOFTWARE = 'HOST_SOFTWARE'  
HOST_VULNERABILITY = 'HOST_VULNERABILITY'
```

`censys.asm.events.format_data` (*start*: *Optional[Union[datetime.datetime, int]] = None*, *filters*: *Optional[List[str]] = None*) → dict
Formats cursor request data into a start date/id and filter list

Parameters

- **start** (*datetime or int, optional*) – Timestamp or event ID to begin searching.
- **filters** (*list, optional*) – List of filters applied to logbook search results.

Returns Formatted logbook cursor request data

Return type dict

censys.asm.seeds module

Interact with the Censys Seeds API.

class `censys.asm.seeds.Seeds` (*api_key*: *Optional[str] = None*, ***kwargs*)

Bases: `censys.asm.api.CensysAsmAPI`

Seeds API class

add_seeds (*seeds*: *list*, *force*: *Optional[bool] = False*) → dict
Add seeds to the ASM platform.

Parameters

- **seeds** (*list*) – List of seed objects to add.
- **force** (*bool, optional*) – Forces replace operation.

Returns Added seeds results.

Return type dict

delete_seed_by_id (*seed_id*: *int*) → dict
Delete a seed in the ASM platform by id.

Parameters **seed_id** (*int*) – Seed ID to delete by.

delete_seeds_by_label (*label*: *str*) → dict
Delete seeds in the ASM platform by label.

Parameters **label** (*str*) – Label name to delete by.

get_seed_by_id (*seed_id: int*) → dict

Requests seed data by ID.

Parameters **seed_id** (*int*) – Seed ID to get.

Returns Seed search result.

Return type dict

get_seeds (*seed_type: Optional[str] = None*) → dict

Requests seed data.

Parameters **seed_type** (*str, optional*) – Seed type ['IP_ADDRESS', 'DOMAIN_NAME', 'CIDR', 'ASN'].

Returns Seed search results.

Return type dict

replace_seeds_by_label (*label: str, seeds: list, force: Optional[bool] = False*) → dict

Replace seeds in the ASM platform by label.

Parameters

- **label** (*str*) – Label name to replace by.
- **seeds** (*list*) – List of seed objects to add.
- **force** (*bool, optional*) – Forces replace operation.

Returns Added and removed seeds results.

Return type dict

Module contents

An easy-to-use and lightweight API wrapper for the Censys ASM platform (censys.io).

6.2 Submodules

6.3 censys.api module

Base for interacting with the Censys Search API.

```
class censys.api.CensysSearchAPI (api_id: Optional[str] = None, api_secret: Optional[str]
                                = None, url: Optional[str] = 'https://censys.io/api/v1',
                                **kwargs)
```

Bases: *censys.base.CensysAPIBase*

This class is the base class for the Data, Certificate, IPv4, and Website index.

DEFAULT_URL: str = 'https://censys.io/api/v1'

Default Search API base URL.

INDEX_NAME: Optional[str] = None

Name of Censys Index.

account () → dict

Gets the current account information. Including email and quota.

Returns Account response.

Return type dict

metadata (*query: str*) → dict

Returns metadata of a given search query.

Parameters **query** (*str*) – The query to be executed.

Returns The metadata of the result set returned.

Return type dict

paged_search (*query: str, fields: Optional[List[str]] = None, page: int = 1, flatten: bool = True*) → dict

Searches the given index for all records that match the given query.

Parameters

- **query** (*str*) – The query to be executed.
- **fields** (*Fields, optional*) – Fields to be returned in the result set.
- **page** (*int, optional*) – The page of the result set. Defaults to 1.
- **flatten** (*bool, optional*) – Flattens fields to dot notation. Defaults to True.

Returns The result set returned.

Return type dict

quota () → dict

Gets the current account's query quota.

Returns Quota response.

Return type dict

report (*query: str, field: str, buckets: int = 50*) → dict

Creates a report on the breakdown of the values of a field in a result set. For more details, see our documentation: <https://censys.io/api/v1/docs/report>

Parameters

- **query** (*str*) – The query to be executed.
- **field** (*str*) – The field you are running a breakdown on.
- **buckets** (*int, optional*) – The maximum number of values. Defaults to 50.

Returns The result set returned.

Return type dict

search (*query: str, fields: Optional[List[str]] = None, page: int = 1, max_records: Optional[int] = None, flatten: bool = True*) → Generator[dict, None, None]

Searches the given index for all records that match the given query. For more details, see our documentation: <https://censys.io/api/v1/docs/search>

Parameters

- **query** (*str*) – The query to be executed.
- **fields** (*Fields, optional*) – Fields to be returned in the result set.
- **page** (*int, optional*) – The page of the result set. Defaults to 1.
- **max_records** (*Optional[int], optional*) – The maximum number of records.
- **flatten** (*bool, optional*) – Flattens fields to dot notation. Defaults to True.

Yields *dict* – The result set returned.

view (*document_id: str*) → *dict*

View the current structured data we have on a specific document. For more details, see our documentation:

<https://censys.io/api/v1/docs/view>

Parameters **document_id** (*str*) – The ID of the document you are requesting.

Returns The result set returned.

Return type *dict*

6.4 censys.base module

Base for interacting with the Censys API's.

class `censys.base.CensysAPIBase` (*url: Optional[str] = None, **kwargs*)

Bases: `object`

This is the base class for API queries.

Parameters

- **url** (*str, optional*) – The URL to make API requests.
- **timeout** (*int, optional*) – Timeout for API requests in seconds.
- **user_agent** (*str, optional*) – Override User-Agent string.
- **proxies** (*dict, optional*) – Configure HTTP proxies.

Raises `CensysException` – Base Exception Class for the Censys API.

DEFAULT_MAX_RETRIES: int = 10

Default max number of API retries.

DEFAULT_TIMEOUT: int = 30

Default API timeout.

DEFAULT_USER_AGENT: str = 'censys/1.1.1'

Default API user agent.

6.5 censys.certificates module

Interact with the Censys Search Certificate API.

class `censys.certificates.CensysCertificates` (**args, **kwargs*)

Bases: `censys.api.CensysSearchAPI`

Interacts with the Certificates index.

INDEX_NAME: Optional[str] = 'certificates'

Name of Censys Index.

MAX_PER_BULK_REQUEST = 50

Max number of bulk requests.

bulk (*fingerprints: List[str]*) → *dict*

Requests data in bulk.

Parameters **fingerprints** (*List[str]*) – List of certificate SHA256 fingerprints.

Returns Search results from an API query.

Return type dict

6.6 censys.cli module

Interact with the Censys Search API through the command line.

class `censys.cli.CensysAPISearch` (***kwargs*)

Bases: object

This class searches the Censys API, taking in options from the command line and returning the results to a CSV or JSON file, or to stdout.

Parameters

- **api_id** (*str, optional*) – The API ID provided by Censys.
- **api_secret** (*str, optional*) – The API secret provided by Censys.
- **start_page** (*int, optional*) – Page number to start from. Defaults to 1.
- **max_pages** (*int, optional*) – The maximum number of pages. Defaults to 10.

csv_fields: `List[str] = []`

A list of fields to be used by the CSV writer.

search_certificates (***kwargs*) → List[dict]

A method to search the Certificates data set via the API.

Parameters

- **query** (*str*) – The string search query.
- **fields** (*list, optional*) – The fields that should be returned with a query.
- **overwrite** (*bool, optional*) – Whether to overwrite or append default fields with user fields. Defaults to False.

Returns A list of results from the query.

Return type Results

search_ipv4 (***kwargs*) → List[dict]

A method to search the IPv4 data set via the API.

Parameters

- **query** (*str*) – The string search query.
- **fields** (*list, optional*) – The fields that should be returned with a query.
- **overwrite** (*bool, optional*) – Whether to overwrite or append default fields with user fields. Defaults to False.

Returns A list of results from the query.

Return type Results

search_websites (***kwargs*) → List[dict]

A method to search the Websites (Alexa Top 1M) data set via the API.

Parameters

- **query** (*str*) – The string search query.

- **fields** (*list, optional*) – The fields that should be returned with a query.
- **overwrite** (*bool, optional*) – Whether to overwrite or append default fields with user fields. Defaults to False.

Returns A list of results from the query.

Return type Results

write_file (*results_list: List[dict], file_format: str = 'screen', file_path: Optional[str] = None*) → *bool*
This method just sorts which format will be used to store the results of the query.

Parameters

- **results_list** (*Results*) – A list of results from the API query.
- **file_format** (*str, optional*) – The format of the output.
- **file_path** (*str optional*) – A path to write results to.

Returns True if wrote out successfully.

Return type bool

class `censys.cli.CensysHNRI` (*api_id: str, api_secret: str*)
Bases: object

This class searches the Censys API, check the user's current IP for risks.

Parameters

- **api_id** (*str, optional*) – The API ID provided by Censys.
- **api_secret** (*str, optional*) – The API secret provided by Censys.

HIGH_RISK_DEFINITION: List[str] = ['telnet', 'redis', 'postgres', 'vnc']

MEDIUM_RISK_DEFINITION: List[str] = ['ssh', 'http', 'https']

static get_current_ip () → str
Uses ipify.org to get the current IP address.

Returns IP address.

Return type str

static risks_to_string (*high_risk: list, medium_risk: list*) → str
Risks to printable string.

Parameters

- **high_risk** (*list*) – Lists of high risks.
- **medium_risk** (*list*) – Lists of medium risks.

Raises *CensysCLIException* – No information/risks found.

Returns Printable string for CLI.

Return type str

translate_risk (*protocols: list*) → Tuple[list, list]
Interpret protocols to risks.

Parameters **protocols** (*list*) – List of slash divided ports/protocols.

Returns Lists of high and medium risks.

Return type Tuple[list, list]

view_current_ip_risks () → str
Gets protocol information for the current IP and returns any risks.

Returns Printable

Return type str

`censys.cli.cli_asm_config` (_)
config asm subcommand.

Parameters _ – Argparse Namespace.

`censys.cli.cli_config` (_)
config subcommand.

Parameters _ – Argparse Namespace.

`censys.cli.get_parser` () → argparse.ArgumentParser
Gets ArgumentParser for CLI.

Returns argparse.ArgumentParser

`censys.cli.hnri` (*args*)
hnri subcommand.

Parameters *args* (*Namespace*) – Argparse Namespace.

`censys.cli.main` ()
main cli function

`censys.cli.search` (*args*)
search subcommand.

Parameters *args* (*Namespace*) – Argparse Namespace.

6.7 censys.config module

Interact with the config file.

`censys.config.check_config` (*config*)
Checks config against default config for fields.

Parameters *config* – Configuration to write.

`censys.config.get_config` ()
Reads and returns config.

`censys.config.write_config` (*config*)
Writes config to file.

Parameters *config* – Configuration to write.

6.8 censys.data module

Interact with the Censys Search Data API.

class `censys.data.CensysData` (*api_id: Optional[str] = None, api_secret: Optional[str] = None, url: Optional[str] = 'https://censys.io/api/v1', **kwargs*)

Bases: `censys.api.CensysSearchAPI`

Interacts with the Data index. For more details, see our documentation: <https://censys.io/api/v1/docs/data>

get_series () → dict

Get data on the types of scans we regularly perform (series).

Returns The result set returned.

Return type dict

view_result (*series_id: str, result_id: str*) → dict

View a specific result of a specific series.

Parameters

- **series_id** (*str*) – The ID of the series.
- **result_id** (*str*) – The ID of the result.

Returns The result set returned.

Return type dict

view_series (*series_id: str*) → dict

Get data on a specific series.

Parameters **series_id** (*str*) – The ID of the series.

Returns The result set returned.

Return type dict

6.9 censys.exceptions module

Exceptions for Censys.

exception `censys.exceptions.CensysAPIException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysException`

Base Exception for Censys API's.

exception `censys.exceptions.CensysAppDownForMaintenanceException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the ASM API is down for maintenance.

exception `censys.exceptions.CensysAsmException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAPIException`

Base Exception for the Censys ASM API.

exception `censys.exceptions.CensysAssociatedAssetsThresholdWarningException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the associated asset count is within the warning threshold.

exception `censys.exceptions.CensysCLIException`

Bases: `censys.exceptions.CensysException`

Exception raised when the CLI is passed invalid arguments.

exception `censys.exceptions.CensysCannotCreateTagWithNewColorException` (`status_code:`
`int`,
`mes-`
`sage:`
`str`,
`body:`
`Op-`
`tional[str]`
`=`
`None`,
`const:`
`Op-`
`tional[str]`
`=`
`None`,
`er-`
`ror_code:`
`Op-`
`tional[int]`
`=`
`None`,
`de-`
`tails:`
`Op-`
`tional[str]`
`=`
`None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the specified tag cannot be created with a new color.

```
exception censys.exceptions.CensysCannotRemoveNonExistentSeedsException (status_code:  
                                                                    int,  
                                                                    mes-  
                                                                    sage:  
                                                                    str,  
                                                                    body:  
                                                                    Op-  
                                                                    tional[str]  
                                                                    =  
                                                                    None,  
                                                                    const:  
                                                                    Op-  
                                                                    tional[str]  
                                                                    =  
                                                                    None,  
                                                                    er-  
                                                                    ror_code:  
                                                                    Op-  
                                                                    tional[int]  
                                                                    =  
                                                                    None,  
                                                                    de-  
                                                                    tails:  
                                                                    Op-  
                                                                    tional[str]  
                                                                    =  
                                                                    None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when trying to remove non existent seed nodes.

```
exception censys.exceptions.CensysCannotRemoveNonSeedsException (status_code:  
                                                                    int, message:  
                                                                    str, body:  
                                                                    Optional[str]  
                                                                    = None, const:  
                                                                    Optional[str]  
                                                                    = None, er-  
                                                                    ror_code:  
                                                                    Optional[int]  
                                                                    = None,  
                                                                    details: Optional[str] =  
                                                                    None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when trying to remove non seed nodes.

```
exception censys.exceptions.CensysCertificateNotFoundException (status_code:
    int, message:
    str, body: Op-
    tional[str] =
    None, const:
    Optional[str]
    = None, er-
    ror_code:
    Optional[int] =
    None, details:
    Optional[str] =
    None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the certificate is not found.

```
exception censys.exceptions.CensysCommentNotFoundException (status_code: int,
    message: str, body:
    Optional[str] = None,
    const: Optional[str]
    = None, error_code:
    Optional[int] = None,
    details: Optional[str]
    = None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the requested comment is not found.

```
exception censys.exceptions.CensysDomainNotFoundException (status_code: int,
    message: str, body:
    Optional[str] = None,
    const: Optional[str]
    = None, error_code:
    Optional[int] = None,
    details: Optional[str] =
    None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the domain is not found.

```
exception censys.exceptions.CensysException
```

Bases: Exception

Base Exception for Censys.

```
class censys.exceptions.CensysExceptionMapper
```

Bases: object

Map status code to Exception for the ASM and Search API.

```
ASM_EXCEPTIONS: Dict[int, Type[censys.exceptions.CensysAsmException]] = {10001: <class
    Map of status code to ASM Exception.
```

```
SEARCH_EXCEPTIONS: Dict[int, Type[censys.exceptions.CensysSearchException]] = {401: <
    Map of status code to Search Exception.
```

exception `censys.exceptions.CensysHostNotFoundException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the requested host is not found.

exception `censys.exceptions.CensysInvalidAPIKeyException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the ASM API key is invalid.

exception `censys.exceptions.CensysInvalidAuthTokenException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the auth token is invalid.

exception `censys.exceptions.CensysInvalidColorException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the specified color is invalid.

exception `censys.exceptions.CensysInvalidCommentException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the comment is invalid.

exception `censys.exceptions.CensysInvalidIPv4AddressException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the IPv4 address is invalid.

exception `censys.exceptions.CensysInvalidLogbookCursorException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the logbook cursor is invalid.

exception `censys.exceptions.CensysInvalidPageSizeException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the page size is invalid.

exception `censys.exceptions.CensysInvalidRequestException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the HTTP request is invalid.

exception `censys.exceptions.CensysInvalidSeedDataException` (`status_code: int`,
`message: str`, `body: Optional[str] = None`,
`const: Optional[str] = None`, `error_code: Optional[int] = None`,
`details: Optional[str] = None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the seed data is invalid.

exception `censys.exceptions.CensysInvalidSeedTypeException` (`status_code: int`,
`message: str`, `body: Optional[str] = None`,
`const: Optional[str] = None`, `error_code: Optional[int] = None`,
`details: Optional[str] = None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the seed type is invalid.

exception `censys.exceptions.CensysJSONDecodeException` (`status_code: int`, `message: str`,
`body: Optional[str] = None`, `const: Optional[str] = None`, `error_code: Optional[int] = None`,
`details: Optional[str] = None`)

Bases: `censys.exceptions.CensysSearchException`

Exception raised when the resource requested is not valid JSON.

exception `censys.exceptions.CensysMissingApiKeyException`

Bases: `censys.exceptions.CensysException`

Exception raised when there is no provided ASM API key.

```

exception censys.exceptions.CensysNeedConfirmationToRemoveParentSeedsException (status_code:
    int,
    mes-
    sage:
    str,
    body:
    Op-
    tional[str]
    =
    None,
    const:
    Op-
    tional[str]
    =
    None,
    er-
    ror_code:
    Op-
    tional[int]
    =
    None,
    de-
    tails:
    Op-
    tional[str]
    =
    None)

```

Bases: `censys.exceptions.CensysAsmException`

Exception raised when confirmation is needed to remove seeds with children.

```

exception censys.exceptions.CensysNotASeedException (status_code: int, message: str,
    body: Optional[str] = None,
    const: Optional[str] = None, er-
    ror_code: Optional[int] = None,
    details: Optional[str] = None)

```

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the requested resource is not a seed.

```

exception censys.exceptions.CensysNotFoundException (status_code: int, message: str,
    body: Optional[str] = None,
    const: Optional[str] = None, er-
    ror_code: Optional[int] = None,
    details: Optional[str] = None)

```

Bases: `censys.exceptions.CensysSearchException`

Exception raised when the resource requested is not found.

exception `censys.exceptions.CensysPageNumberOutOfRangeException` (`status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the requested page number is out of range [1 - totalPages].

exception `censys.exceptions.CensysRateLimitExceededException` (`status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None`)

Bases: `censys.exceptions.CensysSearchException`

Exception raised when your Censys rate limit has been exceeded.

exception `censys.exceptions.CensysSearchException` (`status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None`)

Bases: `censys.exceptions.CensysAPIException`

Base Exception for the Censys search API.

exception `censys.exceptions.CensysSeedNotFoundExpection` (`status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the requested seed can not be found.

```

exception censys.exceptions.CensysTagColorHasTrailingOrLeadingWhitespaceException (status_code:
    int,
    message:
    str,
    body:
    Optional[str]
    =
    None,
    const:
    Optional[str]
    =
    None,
    error_code:
    Optional[int]
    =
    None,
    details:
    Optional[str]
    =
    None)

```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the specified tag color has trailing or leading whitespace.

```

exception censys.exceptions.CensysTagColorTooLongException (status_code:    int,
    message:    str, body:
    Optional[str] = None,
    const:    Optional[str]
    = None, error_code:
    Optional[int] = None,
    details:    Optional[str]
    = None)

```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the specified tag color is too long.

```
exception censys.exceptions.CensysTagHasTrailingOrLeadingWhitespaceException (status_code:
                                                                    int,
                                                                    mes-
                                                                    sage:
                                                                    str,
                                                                    body:
                                                                    Op-
                                                                    tional[str]
                                                                    =
                                                                    None,
                                                                    const:
                                                                    Op-
                                                                    tional[str]
                                                                    =
                                                                    None,
                                                                    er-
                                                                    ror_code:
                                                                    Op-
                                                                    tional[int]
                                                                    =
                                                                    None,
                                                                    de-
                                                                    tails:
                                                                    Op-
                                                                    tional[str]
                                                                    =
                                                                    None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the specified tag has trailing or leading whitespace.

```
exception censys.exceptions.CensysTagIsEmptyStringException (status_code:  int,
                                                                    message:  str,
                                                                    body:  Optional[str]
                                                                    = None,  const:
                                                                    Optional[str]  =
                                                                    None,  error_code:
                                                                    Optional[int]  =
                                                                    None,  details:  Op-
                                                                    tional[str] = None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the specified tag is an empty string.

```
exception censys.exceptions.CensysTagLabelTooLongException (status_code:  int,
                                                                    message:  str,  body:
                                                                    Optional[str] = None,
                                                                    const:  Optional[str]
                                                                    = None,  error_code:
                                                                    Optional[int] = None,
                                                                    details:  Optional[str]
                                                                    = None)
```

Bases: *censys.exceptions.CensysAsmException*

Exception raised when the specified tag label is too long.

exception `censys.exceptions.CensysTagLabelsDifferOnlyInCasingException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the specified tag differs from an existing tag in only casing.

exception `censys.exceptions.CensysTooManyInputNodesException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when there are too many input nodes.

exception `censys.exceptions.CensysTooManyRequestsException` (*status_code*: `int`, *message*: `str`, *body*: `Optional[str]` = `None`, *const*: `Optional[str]` = `None`, *error_code*: `Optional[int]` = `None`, *details*: `Optional[str]` = `None`)

Bases: `censys.exceptions.CensysAsmException`

Exception raised when the allowed requests bandwidth is exceeded.

exception `censys.exceptions.CensysUnauthorizedException` (*status_code: int, message: str, body: Optional[str] = None, const: Optional[str] = None, error_code: Optional[int] = None, details: Optional[str] = None*)

Bases: `censys.exceptions.CensysSearchException`

Exception raised when your Censys account doesn't have access to the requested resource.

6.10 censys.ipv4 module

Interact with the Censys Search IPv4 API.

class `censys.ipv4.CensysIPv4` (*api_id: Optional[str] = None, api_secret: Optional[str] = None, url: Optional[str] = 'https://censys.io/api/v1', **kwargs*)

Bases: `censys.api.CensysSearchAPI`

Interacts with the IPv4 index.

INDEX_NAME: Optional[str] = 'ipv4'
Name of Censys Index.

6.11 censys.websites module

Interact with the Censys Search Website API.

class `censys.websites.CensysWebsites` (*api_id: Optional[str] = None, api_secret: Optional[str] = None, url: Optional[str] = 'https://censys.io/api/v1', **kwargs*)

Bases: `censys.api.CensysSearchAPI`

Interacts with the Websites index.

INDEX_NAME: Optional[str] = 'websites'
Name of Censys Index.

DEVELOPMENT

Clone the repository:

with SSH

with HTTPS

```
$ git clone git@github.com:censys/censys-python.git
```

```
$ git clone https://github.com/censys/censys-python.git
```

Install dependencies via pip:

```
$ pip install -e ".[dev]"
```

Run the test suite with `pytest`. More information about testing is available at [Testing](#).

TESTING

Testing is done using `pytest`.

To run the full test suite against your changes, install *dev dependencies* and simply run `pytest`. Which should return without any errors.

```
$ pytest
```

Note: Tests currently require credentials to be setup. More information about credentials is available at [Quick Start](#).

CONTRIBUTING

All contributions (no matter how small) are always welcome.

9.1 Working on your first Pull Request?

You can learn how from this *free* series [How to Contribute to an Open Source Project on GitHub](#)

PYTHON MODULE INDEX

C

- `censys, ??`
- `censys.api, 23`
- `censys.asm, 23`
- `censys.asm.api, 19`
- `censys.asm.assets, 19`
- `censys.asm.client, 21`
- `censys.asm.events, 21`
- `censys.asm.seeds, 22`
- `censys.base, 25`
- `censys.certificates, 25`
- `censys.cli, 26`
- `censys.config, 28`
- `censys.data, 29`
- `censys.exceptions, 29`
- `censys.ipv4, 42`
- `censys.websites, 42`

Symbols

--api-id <api_id>
 censys-hnri command line option, 17
 censys-search command line option, 18
 --api-secret <api_secret>
 censys-hnri command line option, 17
 censys-search command line option, 18
 --fields <fields>
 censys-search command line option, 18
 --format <json|csv|screen>
 censys-search command line option, 18
 --help
 censys command line option, 17
 censys-config command line option, 17
 censys-config-asm command line option, 17
 censys-hnri command line option, 17
 censys-search command line option, 18
 --index-type {ipv4,certs,websites}
 censys-search command line option, 18
 --max-pages <max_pages>
 censys-search command line option, 18
 --output <output>
 censys-search command line option, 18
 --overwrite
 censys-search command line option, 18
 --query <query>
 censys-search command line option, 18
 --start-page <start_page>
 censys-search command line option, 18

-f <json|csv|screen>
 censys-search command line option, 18
 -h
 censys command line option, 17
 censys-config command line option, 17
 censys-config-asm command line option, 17
 censys-hnri command line option, 17
 censys-search command line option, 18
 -o <output>
 censys-search command line option, 18
 -q <query>
 censys-search command line option, 18

A

account () (*censys.api.CensysSearchAPI method*), 23
 add_comment () (*censys.asm.assets.Assets method*), 19
 add_seeds () (*censys.asm.seeds.Seeds method*), 22
 add_tag () (*censys.asm.assets.Assets method*), 19
 ASM_EXCEPTIONS (*censys.exceptions.CensysExceptionMapper attribute*), 33
 AsmClient (*class in censys.asm.client*), 21
 Assets (*class in censys.asm.assets*), 19

B

bulk () (*censys.certificates.CensysCertificates method*), 25

C

censys
 module, 1
 censys command line option
 --help, 17
 -h, 17
 censys.api

- module, 23
- censys.asm
 - module, 23
- censys.asm.api
 - module, 19
- censys.asm.assets
 - module, 19
- censys.asm.client
 - module, 21
- censys.asm.events
 - module, 21
- censys.asm.seeds
 - module, 22
- censys.base
 - module, 25
- censys.certificates
 - module, 25
- censys.cli
 - module, 26
- censys.config
 - module, 28
- censys.data
 - module, 29
- censys.exceptions
 - module, 29
- censys.ipv4
 - module, 42
- censys.websites
 - module, 42
- censys-config command line option
 - help, 17
 - h, 17
- censys-config-asm command line option
 - help, 17
 - h, 17
- censys-hnri command line option
 - api-id <api_id>, 17
 - api-secret <api_secret>, 17
 - help, 17
 - h, 17
- censys-search command line option
 - api-id <api_id>, 18
 - api-secret <api_secret>, 18
 - fields <fields>, 18
 - format <json|csv|screen>, 18
 - help, 18
 - index-type {ipv4,certs,websites}, 18
 - max-pages <max_pages>, 18
 - output <output>, 18
 - overwrite, 18
 - query <query>, 18
 - start-page <start_page>, 18
 - f <json|csv|screen>, 18
 - h, 18
 - o <output>, 18
 - q <query>, 18
- CensysAPIBase (class in censys.base), 25
- CensysAPIException, 29
- CensysAPISearch (class in censys.cli), 26
- CensysAppDownForMaintenanceException, 29
- CensysAsmAPI (class in censys.asm.api), 19
- CensysAsmException, 30
- CensysAssociatedAssetsThresholdWarningException, 30
- CensysCannotCreateTagWithNewColorException, 31
- CensysCannotRemoveNonExistentSeedsException, 31
- CensysCannotRemoveNonSeedsException, 32
- CensysCertificateNotFoundException, 32
- CensysCertificates (class in censys.certificates), 25
- CensysCLIException, 30
- CensysCommentNotFoundException, 33
- CensysData (class in censys.data), 29
- CensysDomainNotFoundException, 33
- CensysException, 33
- CensysExceptionMapper (class in censys.exceptions), 33
- CensysHNRI (class in censys.cli), 27
- CensysHostNotFoundException, 33
- CensysInvalidAPIKeyException, 34
- CensysInvalidAuthTokenException, 34
- CensysInvalidColorException, 34
- CensysInvalidCommentException, 34
- CensysInvalidIPv4AddressException, 34
- CensysInvalidLogbookCursorException, 35
- CensysInvalidPageSizeException, 35
- CensysInvalidRequestException, 35
- CensysInvalidSeedDataException, 35
- CensysInvalidSeedTypeException, 36
- CensysIPv4 (class in censys.ipv4), 42
- CensysJSONDecodeException, 36
- CensysMissingApiKeyException, 36
- CensysNeedConfirmationToRemoveParentSeedsException, 36
- CensysNotASeedException, 37
- CensysNotFoundException, 37
- CensysPageNumberOutOfRangeException, 37
- CensysRateLimitExceededException, 38
- CensysSearchAPI (class in censys.api), 23
- CensysSearchException, 38
- CensysSeedNotFoundException, 38
- CensysTagColorHasTrailingOrLeadingWhitespaceException, 38
- CensysTagColorTooLongException, 39

- CensysTagHasTrailingOrLeadingWhitespaceException, 39
- CensysTagIsEmptyStringException, 40
- CensysTagLabelsDifferOnlyInCasingException, 40
- CensysTagLabelTooLongException, 40
- CensysTooManyInputNodesException, 41
- CensysTooManyRequestsException, 41
- CensysUnauthorizedException, 41
- CensysWebsites (class in *censys.websites*), 42
- CERT (*censys.asm.events.Filters* attribute), 21
- CERT_RISK (*censys.asm.events.Filters* attribute), 21
- check_config() (in module *censys.config*), 28
- cli_asm_config() (in module *censys.cli*), 28
- cli_config() (in module *censys.cli*), 28
- csv_fields (*censys.cli.CensysAPISearch* attribute), 26
- ## D
- DEFAULT_MAX_RETRIES (*censys.base.CensysAPIBase* attribute), 25
- DEFAULT_TIMEOUT (*censys.base.CensysAPIBase* attribute), 25
- DEFAULT_URL (*censys.api.CensysSearchAPI* attribute), 23
- DEFAULT_URL (*censys.asm.api.CensysAsmAPI* attribute), 19
- DEFAULT_USER_AGENT (*censys.base.CensysAPIBase* attribute), 25
- delete_seed_by_id() (*censys.asm.seeds.Seeds* method), 22
- delete_seeds_by_label() (*censys.asm.seeds.Seeds* method), 22
- delete_tag() (*censys.asm.assets.Assets* method), 20
- DOMAIN (*censys.asm.events.Filters* attribute), 21
- DOMAIN_EXPIRATION_DATE (*censys.asm.events.Filters* attribute), 21
- DOMAIN_MAIL_EXCHANGE_SERVER (*censys.asm.events.Filters* attribute), 21
- DOMAIN_NAME_SERVER (*censys.asm.events.Filters* attribute), 21
- DOMAIN_REGISTRAR (*censys.asm.events.Filters* attribute), 21
- DOMAIN_RISK (*censys.asm.events.Filters* attribute), 22
- DOMAIN_SUBDOMAIN (*censys.asm.events.Filters* attribute), 22
- ## E
- Events (class in *censys.asm.events*), 21
- ## F
- Filters (class in *censys.asm.events*), 21
- format_data() (in module *censys.asm.events*), 22
- format_tag() (in module *censys.asm.assets*), 20
- get_asset_by_id() (*censys.asm.assets.Assets* method), 20
- get_assets() (*censys.asm.assets.Assets* method), 20
- get_comment_by_id() (*censys.asm.assets.Assets* method), 20
- get_comments() (*censys.asm.assets.Assets* method), 20
- get_config() (in module *censys.config*), 28
- get_current_ip() (*censys.cli.CensysHNRI* static method), 27
- get_cursor() (*censys.asm.events.Events* method), 21
- get_events() (*censys.asm.events.Events* method), 21
- get_parser() (in module *censys.cli*), 28
- get_seed_by_id() (*censys.asm.seeds.Seeds* method), 22
- get_seeds() (*censys.asm.seeds.Seeds* method), 23
- get_series() (*censys.data.CensysData* method), 29
- ## G
- ## H
- HIGH_RISK_DEFINITION (*censys.cli.CensysHNRI* attribute), 27
- hnri() (in module *censys.cli*), 28
- HOST (*censys.asm.events.Filters* attribute), 22
- HOST_CERT (*censys.asm.events.Filters* attribute), 22
- HOST_PORT (*censys.asm.events.Filters* attribute), 22
- HOST_PROTOCOL (*censys.asm.events.Filters* attribute), 22
- HOST_RISK (*censys.asm.events.Filters* attribute), 22
- HOST_SOFTWARE (*censys.asm.events.Filters* attribute), 22
- HOST_VULNERABILITY (*censys.asm.events.Filters* attribute), 22
- ## I
- INDEX_NAME (*censys.api.CensysSearchAPI* attribute), 23
- INDEX_NAME (*censys.certificates.CensysCertificates* attribute), 25
- INDEX_NAME (*censys.ipv4.CensysIPv4* attribute), 42
- INDEX_NAME (*censys.websites.CensysWebsites* attribute), 42
- ## M
- main() (in module *censys.cli*), 28
- MAX_PER_BULK_REQUEST (*censys.certificates.CensysCertificates* attribute), 25
- MEDIUM_RISK_DEFINITION (*censys.cli.CensysHNRI* attribute), 27
- metadata() (*censys.api.CensysSearchAPI* method), 24
- module
censys, 1

- `censys.api`, 23
- `censys.asm`, 23
- `censys.asm.api`, 19
- `censys.asm.assets`, 19
- `censys.asm.client`, 21
- `censys.asm.events`, 21
- `censys.asm.seeds`, 22
- `censys.base`, 25
- `censys.certificates`, 25
- `censys.cli`, 26
- `censys.config`, 28
- `censys.data`, 29
- `censys.exceptions`, 29
- `censys.ipv4`, 42
- `censys.websites`, 42

P

`paged_search()` (*censys.api.CensysSearchAPI method*), 24

Q

`quota()` (*censys.api.CensysSearchAPI method*), 24

R

`replace_seeds_by_label()` (*censys.asm.seeds.Seeds method*), 23

`report()` (*censys.api.CensysSearchAPI method*), 24

`risks_to_string()` (*censys.cli.CensysHNRI static method*), 27

S

`search()` (*censys.api.CensysSearchAPI method*), 24

`search()` (*in module censys.cli*), 28

`search_certificates()` (*censys.cli.CensysAPISearch method*), 26

`SEARCH_EXCEPTIONS` (*censys.exceptions.CensysExceptionMapper attribute*), 33

`search_ipv4()` (*censys.cli.CensysAPISearch method*), 26

`search_websites()` (*censys.cli.CensysAPISearch method*), 26

`Seeds` (*class in censys.asm.seeds*), 22

T

`translate_risk()` (*censys.cli.CensysHNRI method*), 27

V

`view()` (*censys.api.CensysSearchAPI method*), 25

`view_current_ip_risks()` (*censys.cli.CensysHNRI method*), 27

`view_result()` (*censys.data.CensysData method*), 29

`view_series()` (*censys.data.CensysData method*), 29

W

`write_config()` (*in module censys.config*), 28

`write_file()` (*censys.cli.CensysAPISearch method*), 27