

---

# **Censys Python**

*Release 2.0.0*

**Censys Team**

**May 10, 2021**



## TABLE OF CONTENTS

<b>1 Quick Start</b>	<b>1</b>
<b>2 Usage v1</b>	<b>3</b>
<b>3 Usage v2</b>	<b>7</b>
<b>4 ASM Usage</b>	<b>9</b>
<b>5 Advanced Usage</b>	<b>15</b>
<b>6 CLI Reference</b>	<b>17</b>
<b>7 API Reference</b>	<b>19</b>
<b>8 Development</b>	<b>59</b>
<b>9 Testing</b>	<b>61</b>
<b>10 Contributing</b>	<b>63</b>
<b>11 License</b>	<b>65</b>
<b>Python Module Index</b>	<b>69</b>
<b>Index</b>	<b>71</b>



## 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 V1

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](#). A list of index fields can be found in the [Censys API definitions page](#).

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.

```
from censys.search import CensysIPv4

c = 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.

```
from censys.search import CensysCertificates

c = CensysCertificates()

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

## 2.3 report

Below we show an example using the CensysWebsites index.

```
from censys.search import CensysWebsites

c = 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.

```
from censys.search import CensysData

c = 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.

```
from censys.search import CensysIPv4

c = 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.

```
from censys.search import CensysCertificates

c = CensysCertificates()

fingerprints = [
    "fce621c0dc1c666d03d660472f636ce91e66e96460545f0da7eb1a24873e2f70",
    "a762bf68f167f6fbdf2ab00fdefeb8b96f91335ad6b483b482dfd42c179be076"
]

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



## USAGE V2

The Censys Search API provides functionality for interacting with Censys resources such as Hosts.

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

- `search` - Allows searches against the Hosts index using the same search syntax as the [web app](#).
- `view` - Returns the structured data we have about a specific Host, given the resource's natural ID.
- `aggregate` - Allows you to view resources as a spectrum based on attributes of the resource, similar to the [Report Builder page](#) on the web app.

More details about each option can be found in the [Censys API documentation](#). A list of index fields can be found in the [Censys API definitions page](#).

Python class objects must be initialized for each resource index (Hosts).

- `CensysHosts`

### 3.1 search

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

```
from censys.search import CensysHosts

h = CensysHosts()

# Single page of search results
query = h.search("service.service_name: HTTP", per_page=5)
print(query())

# Multiple pages of search results
# You can optionally pass in a number of results to be returned
# each page and the number of pages you want returned.
for page in h.search("service.service_name: HTTP", per_page=5, pages=2):
    print(page)
```

## 3.2 view

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

```
from censys.search import CensysHosts

h = CensysHosts()

# Fetch a specific host and its services
host = h.view("8.8.8.8")
print(host)

# You can optionally pass in a RFC3339 timestamp to
# fetch a host at the given point in time.
# Please note historical API access is required.
host = h.view("8.8.8.8", at_time="2021-03-01T17:49:05Z")
print(host)

# You can also pass in a date or datetime object.
from datetime import date
host = h.view("8.8.8.8", at_time=date(2021, 3, 1))
print(host)
```

## 3.3 aggregate

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

```
from censys.search import CensysHosts

h = CensysHosts()

# The aggregate method constructs a report using a query, an aggregation field, and
# the
# number of buckets to bin.
report = h.aggregate(
    "service.service_name: HTTP",
    "services.port",
    num_buckets=5,
)
print(report)
```

## 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 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 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 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 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 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 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 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 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 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))
```



## ADVANCED USAGE

### 5.1 Proxies

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

```
from censys.search import CensysIPv4

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

c = 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.



## CLI REFERENCE

### 6.1 censys

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

- h, --help**  
show this help message and exit
- v, --version**  
display version

#### 6.1.1 censys config

Configure Censys Search API Settings

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

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

#### 6.1.2 censys config-asm

Configure Censys ASM API Settings

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

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

#### 6.1.3 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)

## 6.1.4 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

### 7.1 censys.asm package

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

**class** `censys.asm.AsmClient` (*api\_key: Optional[str] = None, \*\*kwargs*)  
Bases: `object`

Client ASM API class.

#### Parameters

- **api\_key** (*str*) – Optional; The API Key provided by Censys.
- **\*\*kwargs** – Arbitrary keyword arguments.

**class** `censys.asm.Assets` (*asset\_type: str, \*args, \*\*kwargs*)  
Bases: `censys.asm.api.CensysAsmAPI`

Assets API class.

#### Parameters

- **asset\_type** (*str*) – Type of asset to interact with.
- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**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.

**Returns** Added tag results.

**Return type** `dict`

**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.

**Returns** Deleted tag results.

**Return type** `dict`

**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*) → `Iterator[dict]`

Requests assets data.

**Parameters**

- **page\_number** (*int*) – Optional; Page number to begin at when searching.
- **page\_size** (*int*) – Optional; Page size for retrieving assets.

**Yields** `dict` – The assets result returned.

**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*) → `Iterator[dict]`

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

**class** `censys.asm.CertificatesAssets` (\*args, \*\*kwargs)

Bases: `censys.asm.assets.assets.Assets`

Certificates Assets API class.



**Parameters**

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**class** `censys.asm.Clouds` (*api\_key: Optional[str] = None, \*\*kwargs*)

Bases: `censys.asm.api.CensysAsmAPI`

Clouds API class.

**base\_path** = 'clouds'

**get\_domain\_counts** (*since: Union[str, datetime.date, datetime.datetime]*) → dict

Retrieve domain counts by cloud.

**Parameters** **since** (*Since*) – Date to include domains from.

**Returns** Domain count result.

**Return type** dict

**get\_host\_counts** (*since: Union[str, datetime.date, datetime.datetime]*) → dict

Retrieve host counts by cloud.

Hosts found after the date provided in the *since* parameter will be included in the new asset counts.

**Parameters** **since** (*Since*) – Date to include hosts from.

**Returns** Host count result.

**Return type** dict

**get\_object\_store\_counts** (*since: Union[str, datetime.date, datetime.datetime]*) → dict

Retrieve object store counts by cloud.

**Parameters** **since** (*Since*) – Date to include object stores from.

**Returns** Object store count result.

**Return type** dict

**get\_subdomain\_counts** (*since: Union[str, datetime.date, datetime.datetime]*) → dict

Retrieve subdomain counts by cloud.

**Parameters** **since** (*Since*) – Date to include subdomains from.

**Returns** Subdomain count result.

**Return type** dict

**get\_unknown\_counts** () → dict

Retrieve known and unknown counts for hosts by cloud.

**Returns** Unknown count result.

**Return type** dict

**class** `censys.asm.DomainsAssets` (*\*args, \*\*kwargs*)

Bases: `censys.asm.assets.assets.Assets`

Domains Assets API class.

**Parameters**

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**get\_subdomains** (*domain: str, page\_number: int = 1, page\_size: Optional[int] = None*) → Iterator[dict]  
List all subdomains of the parent domain.

**Parameters**

- **domain** – (str): Parent domain to query.
- **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

**class** `censys.asm.Events` (*api\_key: Optional[str] = None, \*\*kwargs*)

Bases: `censys.asm.api.CensysAsmAPI`

Events API class.

**base\_path** = 'logbook'

**get\_cursor** (*start: Optional[Union[datetime.datetime, int]] = None, filters: Optional[List[str]] = None*) → str  
Requests a logbook cursor.

**Parameters**

- **start** (*[datetime.datetime, 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*) → Iterator[dict]  
Requests logbook events from inception or from the provided cursor.

**Parameters** **cursor** (*str*) – Optional; Logbook cursor.

**Yields** *dict* – Logbook event.

**class** `censys.asm.HostsAssets` (*\*args, \*\*kwargs*)

Bases: `censys.asm.assets.assets.Assets`

Hosts Assets API class.

**Parameters**

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**class** `censys.asm.Seeds` (*api\_key: Optional[str] = None, \*\*kwargs*)

Bases: `censys.asm.api.CensysAsmAPI`

Seeds API class.

**add\_seeds** (*seeds: list, force: Optional[bool] = None*) → 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`

`base_path = 'seeds'`

`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.

**Returns** Delete results.

**Return type** `dict`

`delete_seeds_by_label(label: str) → dict`

Delete seeds in the ASM platform by label.

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

**Returns** Delete results.

**Return type** `dict`

`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] = None) → 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`

### 7.1.1 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.common.base.CensysAPIBase`

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

#### Parameters

- **api\_key** (*str*) – Optional; The API Key provided by Censys.
- **\*\*kwargs** – Arbitrary keyword arguments.

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

**DEFAULT\_URL:** `str = 'https://app.censys.io/api/v1'`  
Default ASM API base URL.

## 7.2 censys.common package

Common Code for the Censys Python SDK.

### 7.2.1 censys.common.base module

Base for interacting with the Censys APIs.

**class** `censys.common.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.
- **max\_retries** (*int*) – Optional; Max number of times to retry failed API requests.
- **user\_agent** (*str*) – Optional; Override User-Agent string.
- **proxies** (*dict*) – Optional; Configure HTTP proxies.
- **\*\*kwargs** – Arbitrary keyword arguments.

**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/2.0.0'`  
Default API user agent.

## 7.2.2 censys.common.config module

Interact with the config file.

`censys.common.config.check_config` (*config: configparser.ConfigParser*) → None  
Checks config against default config for fields.

**Parameters** `config` (*configparser.ConfigParser*) – Configuration to write.

**Returns** Config for Censys.

**Return type** `configparser.ConfigParser`

`censys.common.config.get_config` () → `configparser.ConfigParser`  
Reads and returns config.

**Returns** Config for Censys.

**Return type** `configparser.ConfigParser`

`censys.common.config.write_config` (*config: configparser.ConfigParser*) → None  
Writes config to file.

**Parameters** `config` – Configuration to write.

## 7.2.3 censys.common.deprecation module

Warns on deprecated class and functions.

**class** `censys.common.deprecation.DeprecationDecorator` (*message: Optional[str] = None*) =

Bases: `object`

Deprecation Decorator for classes and functions.

**Parameters** `message` (*str*) – Optional; Message to display to user.

## 7.2.4 censys.common.exceptions module

Exceptions for Censys.

**exception** `censys.common.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.common.exceptions.CensysException`

Base Exception for Censys APIs.

**Parameters**

- **status\_code** (*int*) – HTTP status code.
- **message** (*str*) – HTTP message.
- **body** (*str*) – Optional; HTTP body.
- **const** (*str*) – Optional; Constant for manual errors.
- **error\_code** (*int*) – Optional; Error code.

- **details** (*str*) – Optional; Additional details.

```
exception censys.common.exceptions.CensysAppDownForMaintenanceException (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.common.exceptions.CensysAsmException*

Exception raised when the ASM API is down for maintenance.

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

Bases: *censys.common.exceptions.CensysAPIException*

Base Exception for the Censys ASM API.

---

```

exception censys.common.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.common.exceptions.CensysAsmException`

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

```

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

```

Bases: `censys.common.exceptions.CensysAsmException`

Exception raised when a bad JSON string is in the body.

```

exception censys.common.exceptions.CensysCLIException

```

Bases: `censys.common.exceptions.CensysException`

Exception raised when the CLI is passed invalid arguments.

**exception** `censys.common.exceptions.CensysCannotCreateTagWithNewColorException` (*status\_code*: *int*, *message*: *str*, *body*: *Optional[str]* = *None*, *const*: *Optional[str]* = *None*, *error\_code*: *Optional[int]* = *None*, *details*: *Optional[str]* = *None*)

Bases: `censys.common.exceptions.CensysAsmException`

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



---

**exception** `censys.common.exceptions.CensysCannotRemoveNonExistentSeedsException` (*status\_code*: *int*, *message*: *str*, *body*: *Optional[str]* = *None*, *const*: *Optional[str]* = *None*, *error\_code*: *Optional[int]* = *None*, *details*: *Optional[str]* = *None*)

Bases: `censys.common.exceptions.CensysAsmException`

Exception raised when trying to remove non existent seed nodes.

```
exception censys.common.exceptions.CensysCannotRemoveNonSeedsException (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.common.exceptions.CensysAsmException*

Exception raised when trying to remove non seed nodes.

---

```

exception censys.common.exceptions.CensysCertificateNotFoundException (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.common.exceptions.CensysAsmException*

Exception raised when the certificate is not found.

```

exception censys.common.exceptions.CensysCommentNotFoundException (status_code:
                                                                    int, mes-
                                                                    sage: str,
                                                                    body: Op-
                                                                    tional[str]
                                                                    = None,
                                                                    const: Op-
                                                                    tional[str]
                                                                    = None,
                                                                    error_code:
                                                                    Op-
                                                                    tional[int]
                                                                    = None,
                                                                    details: Op-
                                                                    tional[str]
                                                                    = None)

```

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the requested comment is not found.

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the domain is not found.

```
exception censys.common.exceptions.CensysException  
Bases: Exception
```

Base Exception for Censys.

```
class censys.common.exceptions.CensysExceptionMapper  
Bases: object
```

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

```
ASM_EXCEPTIONS: Dict [int, Type [censys.common.exceptions.CensysAsmException]] = {10000:  
    Map of status code to ASM Exception.
```

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

```
exception censys.common.exceptions.CensysHostNotFoundException (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.common.exceptions.CensysAsmException*

Exception raised when the requested host is not found.

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the ASM API key is invalid.

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the auth token is invalid.

```
exception censys.common.exceptions.CensysInvalidCloudAssetDataException (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.common.exceptions.CensysAsmException*

Exception raised when invalid cloud asset data is submitted.

```
exception censys.common.exceptions.CensysInvalidCloudException (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.common.exceptions.CensysAsmException*

Exception raised when an invalid cloud is submitted.

```
exception censys.common.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.common.exceptions.CensysAsmException*

Exception raised when the specified color is invalid.

```
exception censys.common.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.common.exceptions.CensysAsmException*

Exception raised when the comment is invalid.

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when an invalid date is submitted.

```
exception censys.common.exceptions.CensysInvalidIPv4AddressException (status_code:  
    int,  
    mes-  
    sage:  
    str,  
    body:  
    Op-  
    tional[str]  
    =  
    None,  
    const:  
    Op-  
    tional[str]  
    =  
    None,  
    er-  
    ror_code:  
    Op-  
    tional[int]  
    =  
    None,  
    details:  
    Op-  
    tional[str]  
    =  
    None)
```

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the IPv4 address is invalid.



---

```
exception censys.common.exceptions.CensysInvalidLogbookCursorException (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.common.exceptions.CensysAsmException*

Exception raised when the logbook cursor is invalid.

**exception** `censys.common.exceptions.CensysInvalidObjectStorageAssetIdentifierException` (*status\_*  
*int,*  
*mes-*  
*sage:*  
*str,*  
*body:*  
*Op-*  
*tional[*  
*=*  
*None,*  
*const:*  
*Op-*  
*tional[*  
*=*  
*None,*  
*er-*  
*ror\_co*  
*Op-*  
*tional[*  
*=*  
*None,*  
*de-*  
*tails:*  
*Op-*  
*tional[*  
*=*  
*None)*

Bases: `censys.common.exceptions.CensysAsmException`

Exception raised when object storage name is not a valid asset URL.

---

```

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

```

Bases: `censys.common.exceptions.CensysAsmException`

Exception raised when no object storage assets with given URL were found.

```

exception censys.common.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.common.exceptions.CensysAsmException`

Exception raised when the page size is invalid.

```
exception censys.common.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.common.exceptions.CensysAsmException*

Exception raised when the HTTP request is invalid.

```
exception censys.common.exceptions.CensysInvalidSeedDataException (status_code:  
int, mes-  
sage: str,  
body: Op-  
tional[str]  
= None,  
const: Op-  
tional[str]  
= None,  
error_code:  
Op-  
tional[int]  
= None,  
details: Op-  
tional[str]  
= None)
```

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the seed data is invalid.

```
exception censys.common.exceptions.CensysInvalidSeedTypeException (status_code:  
int, mes-  
sage: str,  
body: Op-  
tional[str]  
= None,  
const: Op-  
tional[str]  
= None,  
error_code:  
Op-  
tional[int]  
= None,  
details: Op-  
tional[str]  
= None)
```

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the seed type is invalid.

---

```
exception censys.common.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.common.exceptions.CensysSearchException*

Exception raised when the resource requested is not valid JSON.

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when there is no provided ASM API key.

```
exception censys.common.exceptions.CensysNeedConfirmationToRemoveParentSeedsException (status_
int,
mes-
sage:
str,
body:
Optional[
=
None,
const:
Optional[
=
None,
er-
ror_co
Optional[
=
None,
de-
tails:
Optional[
=
None)
```

Bases: *censys.common.exceptions.CensysAsmException*

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

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the requested resource is not a seed.

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

Bases: *censys.common.exceptions.CensysSearchException*

Exception raised when the resource requested is not found.

---

```

exception censys.common.exceptions.CensysPageNumberOutOfRangeException (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.common.exceptions.CensysAsmException*

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

```

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

```

Bases: *censys.common.exceptions.CensysSearchException*

Exception raised when your Censys rate limit has been exceeded.

```
exception censys.common.exceptions.CensysRiskNotFoundException (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.common.exceptions.CensysAsmException*

Exception raised when no risks are found with given risk\_id.

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

Bases: *censys.common.exceptions.CensysAPIException*

Base Exception for the Censys search API.

```
exception censys.common.exceptions.CensysSeedNotFoundException (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.common.exceptions.CensysAsmException*

Exception raised when the requested seed can not be found.



---

**exception** `censys.common.exceptions.CensysTagColorHasTrailingOrLeadingWhitespaceException` (*sta*

*in*  
*me*  
*sa*  
*st*  
*ba*  
*O,*  
*tic*  
*=*  
*Ne*  
*co*  
*O,*  
*tic*  
*=*  
*Ne*  
*er*  
*ro*  
*O,*  
*tic*  
*=*  
*Ne*  
*de*  
*ta*  
*O,*  
*tic*  
*=*  
*Ne*

Bases: `censys.common.exceptions.CensysAsmException`

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

**exception** `censys.common.exceptions.CensysTagColorTooLongException` (*status\_code:*

*int, mes-*  
*sage: str,*  
*body: Op-*  
*tional[str]*  
*= None,*  
*const: Op-*  
*tional[str]*  
*= None,*  
*error\_code:*  
*Op-*  
*tional[int]*  
*= None,*  
*details: Op-*  
*tional[str]*  
*= None)*

Bases: `censys.common.exceptions.CensysAsmException`

Exception raised when the specified tag color is too long.

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

Bases: *censys.common.exceptions.CensysAsmException*

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

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the specified tag is an empty string.

**exception** `censys.common.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.common.exceptions.CensysAsmException`

Exception raised when the specified tag label is too long.

**exception** `censys.common.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.common.exceptions.CensysAsmException`

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

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

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when there are too many input nodes.

```
exception censys.common.exceptions.CensysTooManyRequestsException (status_code:
                                                                    int, mes-
                                                                    sage: str,
                                                                    body: Op-
                                                                    tional[str]
                                                                    = None,
                                                                    const: Op-
                                                                    tional[str]
                                                                    = None,
                                                                    error_code:
                                                                    Op-
                                                                    tional[int]
                                                                    = None,
                                                                    details: Op-
                                                                    tional[str]
                                                                    = None)
```

Bases: *censys.common.exceptions.CensysAsmException*

Exception raised when the allowed requests bandwidth is exceeded.

```
exception censys.common.exceptions.CensysUnauthorizedException (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.common.exceptions.CensysSearchException`

Exception raised when you doesn't have access to the requested resource.

## 7.3 censys.search package

### 7.3.1 censys.search.v1 package

Interact with the Censys Search v1 APIs.

**class** `censys.search.v1.CensysCertificates` (`api_id: Optional[str] = None, api_secret: Optional[str] = None, **kwargs`)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

Interacts with the Certificates index.

See `CensysSearchAPIv1` for additional arguments.

#### Parameters

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**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 bulk certificates.

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

**Returns** Search results from an API query.

**Return type** dict

**bulk\_path = '/bulk/certificates'**

**class** `censys.search.v1.CensysData` (`api_id: Optional[str] = None, api_secret: Optional[str] = None, **kwargs`)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

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`

**class** `censys.search.v1.CensysIPv4` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

Interacts with the IPv4 index.

**INDEX\_NAME: Optional[str] = 'ipv4'**

Name of Censys Index.

**class** `censys.search.v1.CensysWebsites` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

Interacts with the Websites index.

**INDEX\_NAME: Optional[str] = 'websites'**

Name of Censys Index.

## **censys.search.v1.api module**

Base for interacting with the Censys Search API.

`censys.search.v1.api.CensysSearchAPI`

alias of `censys.search.v1.api.CensysSearchAPIv1`

**class** `censys.search.v1.api.CensysSearchAPIv1` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.common.base.CensysAPIBase`

This class is the base class for all v1 API indexes.

See `CensysAPIBase` for additional arguments.

### **Parameters**

- **api\_id** (*str*) – Optional; The API ID provided by Censys.
- **api\_secret** (*str*) – Optional; The API secret provided by Censys.
- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

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

**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.

This includes 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.

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

**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) → Iterator[dict]

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** (*int*) – Optional; The maximum number of records.
- **flatten** (*bool*) – Optional; Flattens fields to dot notation. Defaults to True.

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

**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*

### 7.3.2 censys.search.v2 package

Interact with the Censys Search v2 APIs.

```
class censys.search.v2.CensysHosts (api_id: Optional[str] = None, api_secret: Optional[str] = None, **kwargs)
```

Bases: *censys.search.v2.api.CensysSearchAPIv2*

Interacts with the Hosts index.

#### Examples

Initiates Censys Hosts.

```
>>> from censys.search import CensysHosts
>>> h = CensysHosts()
```

Simple host search.

```
>>> for page in h.search("service.service_name: HTTP"):
>>>     print(page)
[
  {
    'services':
      [
        {'service_name': 'HTTP', 'port': 80},
        {'service_name': 'HTTP', 'port': 443}
      ],
    'ip': '1.0.0.0'
  },
  ...
]
```

View specific host.



```
>>> h.view("1.0.0.0")
{
  'ip': '8.8.8.8',
  'services': [{}],
  ...
}
```

Simple host aggregate.

```
>>> h.aggregate("service.service_name: HTTP", "services.port", num_buckets=5)
{
  'total_omitted': 591527370,
  'buckets': [
    {'count': 56104072, 'key': '80'},
    {'count': 43527894, 'key': '443'},
    {'count': 23070429, 'key': '7547'},
    {'count': 12970769, 'key': '30005'},
    {'count': 12825150, 'key': '22'}
  ],
  'potential_deviation': 3985101,
  'field': 'services.port',
  'query': 'service.service_name: HTTP',
  'total': 172588754
}
```

**INDEX\_NAME:** `str = 'hosts'`  
Name of Censys Index.

### censys.search.v2.api module

Base for interacting with the Censys Search API.

**class** `censys.search.v2.api.CensysSearchAPIv2` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.common.base.CensysAPIBase`

This class is the base class for the Hosts index.

See `CensysAPIBase` for additional arguments.

#### Parameters

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

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

## Examples

```
>>> c = CensysSearchAPIv2()
```

**DEFAULT\_URL:** `str = 'https://search.censys.io/api/v2'`  
Default Search API base URL.

**INDEX\_NAME:** `str = ''`  
Name of Censys Index.

**class Query** (*api: censys.search.v2.api.CensysSearchAPIv2, query: str, per\_page: Optional[int] = None, cursor: Optional[str] = None, pages: int = 1*)  
Bases: Iterable

Query class that is callable and iterable.

Object Searches the given index for all records that match the given query. For more details, see our documentation: <https://search.censys.io/api/v2/docs>

### Parameters

- **api** (*CensysSearchAPIv2*) – Parent API object.
- **query** (*str*) – The query to be executed.
- **per\_page** (*int*) – Optional; The number of results to be returned for each page. Defaults to 100.
- **cursor** (*int*) – Optional; The cursor of the desired result set.
- **pages** (*int*) – Optional; The number of pages returned. Defaults to 1.

**aggregate** (*query: str, field: str, num\_buckets: Optional[int] = None*) → dict  
Aggregate current index.

Creates a report on the breakdown of the values of a field in a result set. For more details, see our documentation: <https://search.censys.io/api/v2/docs>

### Parameters

- **query** (*str*) – The query to be executed.
- **field** (*str*) – The field you are running a breakdown on.
- **num\_buckets** (*int*) – Optional; The maximum number of values. Defaults to 50.

**Returns** The result set returned.

**Return type** dict

**search** (*query: str, per\_page: Optional[int] = None, cursor: Optional[str] = None, pages: int = 1*) → *censys.search.v2.api.CensysSearchAPIv2.Query*  
Search current index.

Searches the given index for all records that match the given query. For more details, see our documentation: <https://search.censys.io/api/v2/docs>

### Parameters

- **query** (*str*) – The query to be executed.
- **per\_page** (*int*) – Optional; The number of results to be returned for each page. Defaults to 100.
- **cursor** (*int*) – Optional; The cursor of the desired result set.
- **pages** (*int*) – Optional; The number of pages returned. Defaults to 1.

**Returns** Query object that can be a callable or an iterable.

**Return type** *Query*

**view** (*document\_id: str, at\_time: Optional[Union[str, datetime.date, datetime.datetime]] = None*) → *dict*  
View document from current index.

View the current structured data we have on a specific document. For more details, see our documentation: <https://search.censys.io/api/v2/docs>

**Parameters**

- **document\_id** (*str*) – The ID of the document you are requesting.
- **at\_time** (*[str, datetime.date, datetime.datetime]*) – Optional; Fetches a document at a given point in time.

**Returns** The result set returned.

**Return type** *dict*

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

**class** `censys.search.CensysCertificates` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: *censys.search.v1.api.CensysSearchAPIv1*

Interacts with the Certificates index.

See `CensysSearchAPIv1` for additional arguments.

**Parameters**

- **\*args** – Variable length argument list.
- **\*\*kwargs** – Arbitrary keyword arguments.

**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 bulk certificates.

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

**Returns** Search results from an API query.

**Return type** *dict*

**bulk\_path = '/bulk/certificates'**

**class** `censys.search.CensysData` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: *censys.search.v1.api.CensysSearchAPIv1*

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

**class** censys.search.CensysHosts (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: *censys.search.v2.api.CensysSearchAPIv2*

Interacts with the Hosts index.

## Examples

Initiates Censys Hosts.

```
>>> from censys.search import CensysHosts
>>> h = CensysHosts()
```

Simple host search.

```
>>> for page in h.search("service.service_name: HTTP"):
>>>     print(page)
[
  {
    'services':
      [
        {'service_name': 'HTTP', 'port': 80},
        {'service_name': 'HTTP', 'port': 443}
      ],
    'ip': '1.0.0.0'
  },
  ...
]
```

View specific host.

```
>>> h.view("1.0.0.0")
{
  'ip': '8.8.8.8',
  'services': [{}],
  ...
}
```

Simple host aggregate.

```
>>> h.aggregate("service.service_name: HTTP", "services.port", num_buckets=5)
{
  'total_omitted': 591527370,
  'buckets': [
    {'count': 56104072, 'key': '80'},
    {'count': 43527894, 'key': '443'},
    {'count': 23070429, 'key': '7547'},
    {'count': 12970769, 'key': '30005'},
    {'count': 12825150, 'key': '22'}
  ],
  'potential_deviation': 3985101,
  'field': 'services.port',
  'query': 'service.service_name: HTTP',
  'total': 172588754
}
```

**INDEX\_NAME:** `str` = `'hosts'`  
Name of Censys Index.

**class** `censys.search.CensysIPv4` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

Interacts with the IPv4 index.

**INDEX\_NAME:** `Optional[str]` = `'ipv4'`  
Name of Censys Index.

**class** `censys.search.CensysWebsites` (*api\_id: Optional[str] = None, api\_secret: Optional[str] = None, \*\*kwargs*)

Bases: `censys.search.v1.api.CensysSearchAPIv1`

Interacts with the Websites index.

**INDEX\_NAME:** `Optional[str]` = `'websites'`  
Name of Censys Index.

**class** `censys.search.SearchClient` (*\*args, \*\*kwargs*)

Bases: `object`

Client for interacting with all Search APIs.

All indexes are passed the args and kwargs that are provided.

## Examples

Inits SearchClient.

```
>>> from censys.search import SearchClient
>>> c = SearchClient()
```

Access both v1 and v2 indexes.

```
>>> certs = c.v1.certificates # CensysCertificates()
>>> data = c.v1.data # CensysData()
>>> ipv4 = c.v1.ipv4 # CensysIPv4()
>>> websites = c.v1.websites # CensysWebsites()
>>> hosts = c.v2.hosts # CensysHosts()
```



## DEVELOPMENT

Clone the repository:

SSH

HTTPS

GitHub CLI

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

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

```
$ gh repo clone censys/censys-python
```

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.

### 10.1 Working on your first Pull Request?

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



Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or “Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its

representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS





## PYTHON MODULE INDEX

### C

- `censys`, ??
- `censys.asm`, 19
- `censys.asm.api`, 24
- `censys.common`, 24
- `censys.common.base`, 24
- `censys.common.config`, 25
- `censys.common.deprecation`, 25
- `censys.common.exceptions`, 25
- `censys.search`, 55
- `censys.search.v1`, 49
- `censys.search.v1.api`, 50
- `censys.search.v2`, 52
- `censys.search.v2.api`, 53



## 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

--version  
     censys command line option, 17  
 -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  
 -v  
     censys command line option, 17

## A

account() (*censys.search.v1.api.CensysSearchAPIv1 method*), 50  
 add\_comment() (*censys.asm.Assets method*), 19  
 add\_seeds() (*censys.asm.Seeds method*), 22  
 add\_tag() (*censys.asm.Assets method*), 19  
 aggregate() (*censys.search.v2.api.CensysSearchAPIv2 method*), 54  
 ASM\_EXCEPTIONS (*censys.common.exceptions.CensysExceptionMapper attribute*), 32  
 AsmClient (*class in censys.asm*), 19  
 Assets (*class in censys.asm*), 19

## B

base\_path (*censys.asm.Clouds attribute*), 21  
 base\_path (*censys.asm.Events attribute*), 22  
 base\_path (*censys.asm.Seeds attribute*), 23  
 bulk() (*censys.search.CensysCertificates method*), 55

`bulk()` (*censys.search.v1.CensysCertificates* method), 49

`bulk_path` (*censys.search.CensysCertificates* attribute), 55

`bulk_path` (*censys.search.v1.CensysCertificates* attribute), 49

## C

`censys`

module, 1

`censys` command line option

--help, 17

--version, 17

-h, 17

-v, 17

`censys.asm`

module, 19

`censys.asm.api`

module, 24

`censys.common`

module, 24

`censys.common.base`

module, 24

`censys.common.config`

module, 25

`censys.common.deprecation`

module, 25

`censys.common.exceptions`

module, 25

`censys.search`

module, 55

`censys.search.v1`

module, 49

`censys.search.v1.api`

module, 50

`censys.search.v2`

module, 52

`censys.search.v2.api`

module, 53

`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.common.base*), 24

`CensysAPIException`, 25

`CensysAppDownForMaintenanceException`, 26

`CensysAsmAPI` (class in *censys.asm.api*), 24

`CensysAsmException`, 26

`CensysAssociatedAssetsThresholdWarningException`, 26

`CensysBadJSONBodyException`, 27

`CensysCannotCreateTagWithNewColorException`, 27

`CensysCannotRemoveNonExistentSeedsException`, 28

`CensysCannotRemoveNonSeedsException`, 29

`CensysCertificateNotFoundException`, 30

`CensysCertificates` (class in *censys.search*), 55

`CensysCertificates` (class in *censys.search.v1*), 49

`CensysCLIException`, 27

`CensysCommentNotFoundException`, 31

`CensysData` (class in *censys.search*), 55

`CensysData` (class in *censys.search.v1*), 49

`CensysDomainNotFoundException`, 31

`CensysException`, 32

`CensysExceptionMapper` (class in *censys.common.exceptions*), 32

`CensysHostNotFoundException`, 32

`CensysHosts` (class in *censys.search*), 56

`CensysHosts` (class in *censys.search.v2*), 52

`CensysInvalidAPIKeyException`, 32

`CensysInvalidAuthTokenException`, 33

`CensysInvalidCloudAssetDataException`, 33

`CensysInvalidCloudException`, 34

`CensysInvalidColorException`, 34

`CensysInvalidCommentException`, 35

`CensysInvalidDateException`, 35

`CensysInvalidIPv4AddressException`, 35

`CensysInvalidLogbookCursorException`, 36

`CensysInvalidObjectStorageAssetIdentifierException`, 37

- CensysInvalidObjectStorageAssetNotFoundException, 38
- CensysInvalidPageSizeException, 39
- CensysInvalidRequestException, 39
- CensysInvalidSeedDataException, 40
- CensysInvalidSeedTypeException, 40
- CensysIPv4 (class in *censys.search*), 57
- CensysIPv4 (class in *censys.search.v1*), 50
- CensysJSONDecodeException, 40
- CensysMissingApiKeyException, 41
- CensysNeedConfirmationToRemoveParentSeedsException, 41
- CensysNotASeedException, 42
- CensysNotFoundException, 42
- CensysPageNumberOutOfRangeException, 42
- CensysRateLimitExceededException, 43
- CensysRiskNotFoundException, 43
- CensysSearchAPI (in module *censys.search.v1.api*), 50
- CensysSearchAPIv1 (class in *censys.search.v1.api*), 50
- CensysSearchAPIv2 (class in *censys.search.v2.api*), 53
- CensysSearchAPIv2.Query (class in *censys.search.v2.api*), 54
- CensysSearchException, 44
- CensysSeedNotFoundException, 44
- CensysTagColorHasTrailingOrLeadingWhitespacesException, 44
- CensysTagColorTooLongException, 45
- CensysTagHasTrailingOrLeadingWhitespacesException, 45
- CensysTagIsEmptyStringException, 46
- CensysTagLabelsDifferOnlyInCasingException, 47
- CensysTagLabelTooLongException, 46
- CensysTooManyInputNodesException, 47
- CensysTooManyRequestsException, 48
- CensysUnauthorizedException, 48
- CensysWebsites (class in *censys.search*), 57
- CensysWebsites (class in *censys.search.v1*), 50
- CertificatesAssets (class in *censys.asm*), 20
- check\_config() (in module *censys.common.config*), 25
- Clouds (class in *censys.asm*), 21
- D**
- DEFAULT\_MAX\_RETRIES (censys.common.base.CensysAPIBase attribute), 24
- DEFAULT\_TIMEOUT (censys.common.base.CensysAPIBase attribute), 24
- DEFAULT\_URL (censys.asm.api.CensysAsmAPI attribute), 24
- DEFAULT\_URL (censys.search.v1.api.CensysSearchAPIv1 attribute), 50
- DEFAULT\_URL (censys.search.v2.api.CensysSearchAPIv2 attribute), 54
- DEFAULT\_USER\_AGENT (censys.common.base.CensysAPIBase attribute), 24
- delete\_seed\_by\_id() (censys.asm.Seeds method), 23
- delete\_seeds\_by\_label() (censys.asm.Seeds method), 23
- delete\_tag() (censys.asm.Assets method), 20
- DeprecationDecorator (class in *censys.common.deprecation*), 25
- DomainsAssets (class in *censys.asm*), 21
- E**
- Events (class in *censys.asm*), 22
- G**
- get\_asset\_by\_id() (censys.asm.Assets method), 20
- get\_assets() (censys.asm.Assets method), 20
- get\_comment\_by\_id() (censys.asm.Assets method), 20
- get\_comments() (censys.asm.Assets method), 20
- get\_config() (in module *censys.common.config*), 25
- get\_cursor() (censys.asm.Events method), 22
- get\_domain\_counts() (censys.asm.Clouds method), 21
- get\_events() (censys.asm.Events method), 22
- get\_host\_counts() (censys.asm.Clouds method), 21
- get\_object\_store\_counts() (censys.asm.Clouds method), 21
- get\_seed\_by\_id() (censys.asm.Seeds method), 23
- get\_seeds() (censys.asm.Seeds method), 23
- get\_series() (censys.search.CensysData method), 55
- get\_series() (censys.search.v1.CensysData method), 49
- get\_subdomain\_counts() (censys.asm.Clouds method), 21
- get\_subdomains() (censys.asm.DomainsAssets method), 21
- get\_unknown\_counts() (censys.asm.Clouds method), 21
- H**
- HostsAssets (class in *censys.asm*), 22

## I

INDEX\_NAME (*censys.search.CensysCertificates* attribute), 55  
 INDEX\_NAME (*censys.search.CensysHosts* attribute), 57  
 INDEX\_NAME (*censys.search.CensysIPv4* attribute), 57  
 INDEX\_NAME (*censys.search.CensysWebsites* attribute), 57  
 INDEX\_NAME (*censys.search.v1.api.CensysSearchAPIv1* attribute), 50  
 INDEX\_NAME (*censys.search.v1.CensysCertificates* attribute), 49  
 INDEX\_NAME (*censys.search.v1.CensysIPv4* attribute), 50  
 INDEX\_NAME (*censys.search.v1.CensysWebsites* attribute), 50  
 INDEX\_NAME (*censys.search.v2.api.CensysSearchAPIv2* attribute), 54  
 INDEX\_NAME (*censys.search.v2.CensysHosts* attribute), 53

## M

MAX\_PER\_BULK\_REQUEST (*censys.search.CensysCertificates* attribute), 55  
 MAX\_PER\_BULK\_REQUEST (*censys.search.v1.CensysCertificates* attribute), 49  
 metadata() (*censys.search.v1.api.CensysSearchAPIv1* method), 51  
 module  
   *censys*, 1  
   *censys.asm*, 19  
   *censys.asm.api*, 24  
   *censys.common*, 24  
   *censys.common.base*, 24  
   *censys.common.config*, 25  
   *censys.common.deprecation*, 25  
   *censys.common.exceptions*, 25  
   *censys.search*, 55  
   *censys.search.v1*, 49  
   *censys.search.v1.api*, 50  
   *censys.search.v2*, 52  
   *censys.search.v2.api*, 53

## P

paged\_search() (*censys.search.v1.api.CensysSearchAPIv1* method), 51

## Q

quota() (*censys.search.v1.api.CensysSearchAPIv1* method), 51

## R

replace\_seeds\_by\_label() (*censys.asm.Seeds* method), 23  
 report() (*censys.search.v1.api.CensysSearchAPIv1* method), 51

## S

search() (*censys.search.v1.api.CensysSearchAPIv1* method), 51  
 search() (*censys.search.v2.api.CensysSearchAPIv2* method), 54  
 SEARCH\_EXCEPTIONS (*censys.common.exceptions.CensysExceptionMapper* attribute), 32  
 SearchClient (class in *censys.search*), 57  
 Seeds (class in *censys.asm*), 22

## V

view() (*censys.search.v1.api.CensysSearchAPIv1* method), 52  
 view() (*censys.search.v2.api.CensysSearchAPIv2* method), 55  
 view\_result() (*censys.search.CensysData* method), 56  
 view\_result() (*censys.search.v1.CensysData* method), 49  
 view\_series() (*censys.search.CensysData* method), 56  
 view\_series() (*censys.search.v1.CensysData* method), 50

## W

write\_config() (in module *censys.common.config*), 25