# Sentinelsat Documentation

***Release 0.6.4***

**Marcel Wille, Kersten Clauss**

September 27, 2016

Contents

Sentinelsat makes finding and downloading Copernicus Sentinel satellite images easy.

It offers an easy to use command line interface.

```
sentinel search --sentinel2 --cloud 30 user password search_polygon.geojson
```

and a powerfull Python API.

```
from sentinelsat.sentinel import SentinelAPI, get_coordinates

api = SentinelAPI('user', 'password')
api.query(
  get_coordinates("search_polygon.geojson"),
  producttype="SLC",
  orbitdirection="ASCENDING"
)
api.download_all()
```

CHAPTER 1

# Contents

## 1.1 Installation

Sentinelsat depends on homura, which depends on PycURL. When the dependencies are fulfilled install with `pip install sentinelsat`.

### 1.1.1 Unix

#### Ubuntu

```
sudo apt-get install build-essential libcurl4-openssl-dev python-dev python-pip
```

#### Fedora

```
sudo yum groupinstall "Development Tools"
sudo yum install libcurl libcurl-devel python-devel python-pip
```

### 1.1.2 Windows

The easiest way to install pycurl is with pycurl wheels provided by Christoph Gohlke

```
pip install pycurl.whl
```

or with‘Conda <http://conda.pydata.org/docs/>‘_

```
conda install pycurl
```

### 1.1.3 OSX

TODO: How to install on OSX.

### 1.1.4 Tests

```
git clone https://github.com/ibamacsr/sentinelsat.git
cd sentinelsat
pip install -e .[test]
export SENTINEL_USER=<your scihub username>
export SENTINEL_PASSWORD=<your scihub password>
py.test -v
```

### 1.1.5 Troubleshooting

The download from Scihub will fail if the server certificate cannot be verified because no default CA bundle is defined, as on Windows, or when the CA bundle is outdated. In most cases the easiest solution is to install or update `certifi`:

`pip install -U certifi` You can also override the the path setting to the PEM file of the CA bundle using the `pass_through_opts` keyword argument when calling `api.download()` or `api.download_all()`:

```
from pycurl import CAINFO
api.download_all(pass_through_opts={CAINFO: 'path/to/my/cacert.pem'})
```

## 1.2 Command Line Interface

Sentinelsat's CLI is divided into two commands:

- `sentinel search` to query and download a number of images over an area
- `sentinel download` to download individual images by their unique identifier

### 1.2.1 Quickstart

A basic search query consists of a search polygon as well as the username and password to access the Scihub.

```
sentinel search [OPTIONS] <user> <password> <geojson>
```

Search areas are provided as GeoJSON polygons, which can be created with QGIS or geojson.io. If you do not specify a start and end date only products published in the last 24 hours will be queried.

Start and end dates refer to the acquisition date given by the *beginPosition <https://scihub.copernicus.eu/userguide/3FullTextSearch>* of the products, i.e. the start of the acquisition time.

#### Sentinel-1

Search and download all Sentinel-1 scenes of type SLC, in descending orbit for the year 2015.

```
sentinel search -s 20150101 -e 20151231 -d \
-q 'producttype=SLC, orbitdirection=Descending' \
-u 'https://scihub.copernicus.eu/dhus' <user> <password> <poly.geojson>
```

Download a single Sentinel-1 GRDH scene covering Santa Claus Village in Finland on Christmas Eve 2015.

```
sentinel download --md5 -u 'https://scihub.copernicus.eu/dhus/' <user> <password> a9048d1d-fea6-4df8-
```

#### Sentinel-2

Search and download Sentinel-2 scenes for January 2016 with a maximum cloud cover of 40%.

```
sentinel search -s 20160101 -e 20160131 --sentinel2 --cloud 40 <user> <password> <poly.geojson>
```

Download all Sentinel-2 scenes published in the last 24 hours.

```
sentinel search --sentinel2 <user> <password> <poly.geojson>
```

### 1.2.2 sentinel search

```
sentinel search [OPTIONS] <user> <password> <geojson>
```

Options:

| | | | |
|---|---|---|---|
| -s | –start | TEXT | Start date of the query in the format YYYYMMDD. |
| -e | –end | TEXT | End date of the query in the format YYYYMMDD. |
| -d | –download | | Download all results of the query. |
| -f | –footprints | | Create geojson file search_footprints.geojson with footprints of the query result. |
| -p | –path | PATH | Set the path where the files will be saved. |
| -q | –query | TEXT | Extra search keywords you want to use in the query. Separate keywords with comma. Example: 'producttype=GRD,polarisationmode=HH'. |
| -u | –url | TEXT | Define another API URL. Default URL is 'https://scihub.copernicus.eu/apihub/'. |
| | –md5 | | Verify the MD5 checksum and write corrupt product ids and filenames to corrupt_scenes.txt. |
| | –sentinel1 | | Limit search to Sentinel-1 products. |
| | –sentinel2 | | Limit search to Sentinel-2 products. |
| -c | –cloud | INT | Maximum cloud cover in percent. (Automatically sets –sentinel2) |
| | –help | | Show help message and exit. |

Query parameters:

ESA maintains a list of valid search keywords to query the SciHub.

### 1.2.3 sentinel download

```
sentinel download [OPTIONS] <user> <password> <productid>
```

Options:

| | | | |
|---|---|---|---|
| -p | –path | PATH | Set the path where the files will be saved. |
| -u | –url | TEXT | Define another API URL. Default URL is 'https://scihub.copernicus.eu/apihub/'. |
| | –md5 | | Verify the MD5 checksum and write corrupt product ids and filenames to corrupt_scenes.txt. |

## 1.3 Python API

### 1.3.1 Quickstart

```
# connect to the API
from sentinelsat.sentinel import SentinelAPI, get_coordinates
api = SentinelAPI('user', 'password', 'https://scihub.copernicus.eu/dhus')

# download single scene by known product id
api.download(<product_id>)

# search by polygon, time, and SciHub query keywords
api.query(get_coordinates(map.geojson), \
          "20151219", date(2015, 12, 29), \
          keywords={"platformname": "Sentinel-2", \
                    "cloudcoverpercentage": "[0 TO 30]"})

# download all results from the search
api.download_all()

# GeoJSON FeatureCollection containing footprints and metadata of the scenes
api.get_footprints()
```

Valid search query keywords can be found at the ESA SciHub documentation.

### 1.3.2 API

**class** sentinelsat.sentinel.**SentinelAPI**(*user*, *password*, *api_url='https://scihub.copernicus.eu/apihub/'*)

Class to connect to Sentinel Data Hub, search and download imagery.

**Parameters** **user** : string

username for DataHub

**password** : string

password for DataHub

**api_url** : string, optional

URL of the DataHub defaults to 'https://scihub.copernicus.eu/apihub'

#### Attributes

| | |
|---|---|
| session | (requests.Session object) Session to connect to DataHub |
| api_url | (str) URL to the DataHub |

#### Methods

| | |
|---|---|
| *download*(id[, path, checksum]) | Download a product using homura's download function. |
| *download_all*([path, checksum]) | Download all products using homura's download function. |
| *format_url*(area[, initial_date, end_date]) | Create the URL to access the SciHub API, defining the max quantity of results to 150 |
| *get_footprints*() | Return the footprints of the resulting scenes in GeoJSON format |

Table 1.1 – continued from previous page

| | |
|---|---|
| *get_product_info*(id) | Access SciHub API to get info about a Product. |
| *get_products*() | Return the result of the Query in json format. |
| *get_products_size*() | Return the total filesize in Gb of all products in the query |
| *query*(area[, initial_date, end_date]) | Query the SciHub API with the coordinates of an area, a date inverval and any other s |

**download**(*id*, *path='.'*, *checksum=False*, ***kwargs*)
Download a product using homura's download function.

If you don't pass the title of the product, it will use the id as filename. Further keyword arguments are passed to the homura.download() function.

**download_all**(*path='.'*, *checksum=False*, ***kwargs*)
Download all products using homura's download function.

It will use the products id as filenames. If the checksum calculation fails a list with tuples of filename and product ids of the corrupt scenes will be returned. Further keyword arguments are passed to the homura.download() function.

**format_url**(*area*, *initial_date=None*, *end_date=datetime.datetime(2016, 9, 27, 8, 58, 27, 302594)*, ***keywords*)
Create the URL to access the SciHub API, defining the max quantity of results to 15000 items.

**get_footprints**()
Return the footprints of the resulting scenes in GeoJSON format

**get_product_info**(*id*)
Access SciHub API to get info about a Product. Returns a dict containing the id, title, size, md5sum, date, footprint and download url of the Product. The date field receives the Start ContentDate of the API.

**get_products**()
Return the result of the Query in json format.

**get_products_size**()
Return the total filesize in Gb of all products in the query

**query**(*area*, *initial_date=None*, *end_date=datetime.datetime(2016, 9, 27, 8, 58, 27, 302563)*, ***keywords*)
Query the SciHub API with the coordinates of an area, a date inverval and any other search keywords accepted by the SciHub API.

sentinelsat.sentinel.**convert_timestamp**(*in_date*)
Convert the timestamp received from Products API, to YYYY-MM-DDThh:mm:ssZ string format.

sentinelsat.sentinel.**format_date**(*in_date*)
Format date or datetime input or a YYYYMMDD string input to YYYY-MM-DDThh:mm:ssZ string format. In case you pass an

sentinelsat.sentinel.**get_coordinates**(*geojson_file*, *feature_number=0*)
Return the coordinates of a polygon of a GeoJSON file.

**Parameters** **geojson_file** : str

location of GeoJSON file_path

**feature_number** : int

Feature to extract polygon from (in case of MultiPolygon FeatureCollection), defaults to first Feature

**Returns** polygon coordinates

string of comma separated coordinate tuples to be used by SentinelAPI

sentinelsat.sentinel.**md5_compare**(*file_path*, *checksum*, *block_size=8192*)
Compare a given md5 checksum with one calculated from a file

## 1.4 Indices and tables

- genindex
- modindex
- search

## S

## C



## D



## F



## G



## M



## Q



## S