
SatNOGS Client Documentation

Release 1.2+0.gdccb527.dirty

SatNOGS

Mar 25, 2020

Contents

1	SatNOGS client architecture	1
1.1	Overview	1
1.2	Modules	1
1.2.1	scheduler	1
1.2.1.1	Tasks	1
1.2.2	observer.observer	2
1.2.3	observer.worker	2
1.2.4	observer.orbital	2
1.2.5	satnogs-client	2
2	satnogsclient Package	3
2.1	satnogsclient Package	3
2.2	settings Module	3
2.3	Subpackages	3
2.3.1	observer Package	3
2.3.1.1	commsocket Module	3
2.3.1.2	observer Module	3
2.3.1.3	orbital Module	3
2.3.1.4	worker Module	3
2.3.2	scheduler Package	3
2.3.2.1	scheduler Package	3
2.3.2.2	tasks Module	3
3	Installation	5

1.1 Overview

SatNOGS client is the part of our software stack that:

- Fetches observation jobs from the network.
- Schedules locally when the observation starts/ends.
- Does orbital calculation for the position of the observer and the tracked object (using `PyEphem`).
- Sends `rotctl/rigctl` commands to control **SatNOGS** rotator.
- Spawns processes to run demodulation/decoding software with the signal received as input.
- Posts observation data back to the network.

1.2 Modules

Following the paradigm of **SatNOGS** being extensively modular, **SatNOGS** client is designed to have discrete modules with specific functionality.

1.2.1 scheduler

- Build using `apscheduler` library.
- Stores tasks in `sqlite`.

1.2.1.1 Tasks

- `get_jobs`: Queries **SatNOGS Network API** to get jobs scheduled for the ground station.
- `spawn_observation`: Initiates an `Observer` instance and runs the observation.

- `post_observation_data`: Gathers output files, parses filename and posts data back to **SatNOGS Network API**.

1.2.2 `observer.observer`

Given initial description of the observation (`tle`, `start`, `end`)

- Checks input for sanity.
- Initializes `WorkerTrack` and `WorkerFreq` instances that start `rigctl/rotctl` communication using `trackstart` method.
- Starts/Stops GNU Radio script (`gr-satnogs`), which collects the data.
- Processes produced data from observation (`ogg` file, waterfall plotting).

1.2.3 `observer.worker`

- Facilitates as a worker for `rigctl/rotctl`.
- Is the lowest abstraction level on `rigctl/rotctl` communications.
- Tracks object until end of observation is reached.

1.2.4 `observer.orbital`

- Implements orbital calculations using `PyEphem`.
- Provides `pinpoint` method that returns `alt/az` position of tracked object.

1.2.5 `satnogs-client`

- `satnogs-client`: A console script which runs the scheduler queue in the background and fetch jobs from the network.

2.1 satnogsclient Package

2.2 settings Module

2.3 Subpackages

2.3.1 observer Package

2.3.1.1 commssocket Module

2.3.1.2 observer Module

2.3.1.3 orbital Module

2.3.1.4 worker Module

2.3.2 scheduler Package

2.3.2.1 scheduler Package

2.3.2.2 tasks Module

Note: These installation steps are intended to be used for contributing to the satnogs-client codebase. If you are interested in setting up satnogs-client for your ground station check the [wiki](#).

Requirements: You will need python, python-virtualenvwrapper, pip and git

1. Install the dependencies

SatNOGS client depends on *libhamlib2* Python bindings. In Debian, these bindings are provided by installing *python3-libhamlib2* package.

2. Build the environment

Clone source code from the [repository](#):

```
$ git clone https://gitlab.com/librespacefoundation/satnogs/satnogs-client.git
```

Set up the virtual environment. On first run you should create it and link it to your project path.:

```
$ cd satnogs-client
$ mkvirtualenv --system-site-packages satnogs-client -a .
```

Activate your python virtual environment:

```
$ workon satnogs-client
```

Install local development requirements:

```
$ pip install -e .
```

3. Run the client

Create an `.env` file on the project root and configure the client environment variables.

Run `satnogs-client`:

```
$ satnogs-client
```