OpenEEW for Python

Release 0.5.0

Dec 04, 2020

Contents

This is the documentation for openeew, a Python package for working with all aspects of the OpenEEW initiative by Grillo.

Currently the package can be used to simplify selecting and downloading OpenEEW data stored as an AWS Public Dataset.

CHAPTER 1

Contents

1.1 Installing

1.1.1 Requirements

Requires Python version 3.5 or later.

1.1.2 Installation

```
pip install openeew
```

1.2 openeew

1.2.1 openeew package

Subpackages

openeew.data package

Submodules

openeew.data.aws module

class openeew.data.aws.**AwsDataClient**(*country_code*, *s3_client=None*)

Bases: object

A client for downloading OpenEEW data stored as an AWS Public Dataset.

Initialize AwsDataClient with the following parameters:

Parameters

- **country_code** (*str*) – The ISO 3166 two-letter country code for which data is required. It is case-insensitive as the value specified will be converted to lower case.
- **s3_client** (*boto3.client.s3*) – The S3 client to use to access OpenEEW data on AWS. If no value is given, an anonymous S3 client will be used.

country_code

Returns ISO 3166 two-letter country code of the client (in lower case). Any data returned by the client will be for this country.

Return type str

get_current_devices()

Gets currently-valid device metadata. Fields are the same as for *get_devices_full_history()*.

Returns A list of device metadata.

Return type list[dict]

get_devices_as_of_date(*date_utc*)

Gets device metadata as of a chosen UTC date. Fields are the same as for *get_devices_full_history()*.

Parameters **date_utc** (*str*) – The UTC date with format %Y-%m-%d %H:%M:%S. E.g. '2018-02-16 23:39:38'.

Returns A list of device metadata.

Return type list[dict]

get_devices_full_history()

Gets full history of device metadata.

Returns A list of device metadata. See Device metadata for information about the fields.

Return type list[dict]

get_filtered_records(*start_date_utc*, *end_date_utc*, *device_ids=None*)

Returns accelerometer records filtered by date and device.

Parameters

- **start_date_utc** (*str*) – The UTC start date with format %Y-%m-%d %H:%M:%S. E.g. '2018-02-16 23:39:38'. Only records with a _RECORD_T equal to or greater than start_date_utc will be returned.
- **end_date_utc** (*str*) – The UTC end date with same format as start_date_utc. Only records with a _RECORD_T equal to or less than end_date_utc will be returned.
- **device_ids** (*Union[str, list[str]]*) – Device IDs that should be returned.

Returns A list of records, where each record is a dict obtained from the stored JSON value. For details about the JSON records, see Data fields for further information about how records are stored.

Return type list[dict]

class openeew.data.aws.**DateTimeKeyBuilder**(*year*, *month=None*, *day=None*, *hour=None*, *minute=None*)

Bases: object

A class for building the datetime part of keys that are organized by a hierarchy that can include year, month, day, hour and minute. The year part of the key must contain the century, i.e. YYYY, and other parts are zero-padded decimals, e.g. 01, 02 etc. An example of such a key template would be "year={}/month={}/day={}/hour={}/{}".

Initialize DateTimeKeyBuilder with the following parameters, which are concatenated together in order of increasing granularity to form key template:

Parameters

- **year** (*str*) – Year part of key template, e.g. "year={}/". Must contain {} for year replacement, where year will be added as YYYY, e.g. 2020.
- **month** (*str*) – Optional month part of key template, e.g. "month={}/". Must contain {} for month replacement, where month will be added as zero-padded number, i.e. 01, 02, . . . , 12.
- **day** (*str*) – Optional day part of key template, e.g. "day={}/". Must contain {} for day replacement, where day will be added as zero-padded number, i.e. 01, 02, . . . , 31.
- **hour** (*str*) – Optional hour part of key template, e.g. "hour={}/". Must contain {} for hour replacement, where hour will be added as zero-padded number, i.e. 00, 01, . . . , 23.
- **minute** (*str*) – Optional minute part of key template, e.g. "{}". Must contain {} for minute replacement, where minute will be added as zero-padded number, i.e. 00, 01, . . . , 59.

get_key_prefixes_within_range(*start_dt*, *end_dt*)

Returns a list of key search prefixes for the given date range, where each prefix corresponds to one day.

Parameters

- **start_dt** (*datetime.datetime*) – The start of the datetime range.
- **end_dt** (*datetime.datetime*) – The end of the datetime range.

Returns List of key search prefixes.

Return type list[str]

get_max_key(*dt*)

Returns maximum possible key value for given datetime.

Parameters **dt** (*datetime.datetime*) – The datetime to build key for.

Returns Key part corresponding to the datetime.

Return type str

get_min_key(*dt*)

Returns minimum possible key value for given datetime.

Parameters **dt** (*datetime.datetime*) – The datetime to build key for.

Returns Key part corresponding to the datetime.

Return type str

template_parts

Returns A list of strings containing the defined template parts, in the following order: year, month, day, hour and minute. The length of list depends on which parts have been specified.

Return type list[str]

openeew.data.df module

openeew.data.df.**get_df_from_records**(*records*, *ref_t_name='cloud_t'*, *ref_axis='x'*)
Returns a pandas DataFrame from a list of records.

Parameters

- **records** (*list[dict]*) – The list of records from which to create a pandas DataFrame.
- **ref_t_name** (*str*) – The name of the time field to use as a reference when calculating sample times. This should be either cloud_t or device_t.
- **ref_axis** (*str*) – The axis to use when determining the number of sample points in each record.

Returns A pandas DataFrame with columns the same as the keys of each record and an additional sample_t column giving an individual timestamp to each of the x, y and z array elements.

Return type pandas.DataFrame

openeew.data.record module

openeew.data.record.**add_sample_t**(*record*, *ref_t_name*, *ref_axis*)
Adds a list of sample times to a record corresponding to each sample point in the record.

Parameters

- **record** (*dict*) – The record to which to add sample times.
- **ref_t_name** (*str*) – The name of the time field to use as a reference when calculating sample times. This should be either cloud_t or device_t.
- **ref_axis** (*str*) – The axis to use when determining the number of sample points in the record.

Returns A record with additional sample_t field containing list of sample times.

Return type dict

openeew.data.record.**add_sample_t_to_records**(*records*, *ref_t_name*, *ref_axis*)
Adds sample_t field to each record in a list of records.

Parameters

- **records** (*list[dict]*) – The list of records to which to add sample times.
- **ref_t_name** (*str*) – The name of the time field to use as a reference when calculating sample times. This should be either cloud_t or device_t.
- **ref_axis** (*str*) – The axis to use when determining the number of sample points in each record.

Returns A list of records with additional sample_t field containing list of sample times.

Return type list[dict]

openeew.data.record.**get_sample_t**(*ref_t*, *idx*, *num_samples*, *sr*)
Calculates Unix time for individual sample point (within a record). The sample point time is calculated by subtracting a multiple of 1/sr from the reference time corresponding to the final sample point in the record, where sr is the sample rate.

Parameters

- **ref_t** (*float*) – The Unix time to use as a basis for the calculation.

- **idx** (*int*) – The zero-based index of the sample point within the record.
- **num_samples** (*int*) – The number of sample points within the record.
- **sr** (*float*) – The sample rate (per second) of the record.

Returns An estimated Unix time corresponding to the sample point.

Return type float

Module contents

This module provides a means to work with OpenEEW data.

Module contents

1.3 License

```
                                 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
```

(continues on next page)

(continued from previous page)

```
      (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
```

(continues on next page)

(continued from previous page)

```
        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.
```

(continues on next page)

(continued from previous page)

```
   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

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright 2019 Grillo Holdings Inc.

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

1.4 Changelog

openeew uses Semantic Versioning

1.4.1 Unreleased

- Allow to get number of sample points in a record using general axis labels
- Pass single device ID as string when retrieving records
- Filter records based on time field name in class attribute
- Create async S3 client based on non-async S3 client metadata #10
- Extract datetime logic to separate class #11

1.4.2 Version 0.5.0

- Use one async client to download all files.

1.4.3 Version 0.4.0

- Use asyncio to speed up downloading of files in AwsDataClient
- Speed up search for S3 keys within specified date range
- Updated some AwsDataClient method decorators

1.4.4 Version 0.3.0

- Added df submodule to openeew.data. The method get_filtered_records_df of AwsDataClient in openeew.data.aws has been removed and instead the function get_df_from_records in openeew.data.df can be used to return a pandas DataFrame from a list of records.
- Fixed/updated docstrings

1.4.5 Version 0.2.0

- Added record submodule to openeew.data

1.4.6 Version 0.1.3

- Fixed/updated docstrings
- Removed Python 3.4 from requirements

1.4.7 Version 0.1.2

- Initial version released on PyPI.

1.5 Contributing

OpenEEW for Python is an open source project and we are always happy to receive contributions from our community. You can contribute in different ways:

- Writing tutorials and blog posts
- Improving the documentation
- Submitting bug reports and feature requests
- Forking this repository and submitting a pull request

1.6 Contributing and Developer information

The community welcomes your involvement and contributions to this project. Please read the OpenEEW [contributing](https://github.com/openeew/openeew/blob/master/CONTRIBUTING.md) document for details on our code of conduct, and the process for submitting pull requests to the community.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

o

Index