hilltop-py Documentation

Release 1.0.0

Mike Kittridge

Oct 26, 2023

SECTIONS

This package contains a variety of Python tools for accessing data contained in Hilltop.

Note: A major rewrite of the web service functions has occurred for version 2, which makes the code much simpler as well as the data requests. Also, there is more consistency with the terminology between Hilltop and hilltop-py as well as within the hilltop-py functions. Consequently, there are some differences that the user will need to pay attention to and update in their scripts when making requests. The new Hilltop module will be the only module being maintained going forward. But the old modules will still be accessible as in the past.

Try it out and provide feedback on the GitHub page to improve the package!

Not all Data Types are currently supported. These include for example HydSection and HydFacecard. The response from the Hilltop server is very different as compared to other Data Types (and I don't really understand what they mean). If there is enough demand (or support), then I might add them.

For further information about the Hilltop server, please look through the official Hilltop server doc.

The GitHub repository is found here.

CHAPTER

ONE

INSTALLATION

Install via pip:

```
pip install hilltop-py
```

Or conda:

```
conda install -c conda-forge hilltop-py
```

1.1 Requirements

The main dependencies are pandas, pydantic, requests, and orjson. pywin32 is a dependency if the user wishes to use the COM module, but it is not installed by default. pywin32 is only for a Windows OS, so the COM module requires the user to be running a Windows OS. This is also likely the case for the native python module.

CHAPTER

TWO

HOW TO USE HILLTOP-PY

This section will describe how to use the hilltop-py package. The functions depend heavily on the Pandas package. Nearly all outputs are Pandas DataFrames.

Note: The API and terminology used in hilltop-py attempts to match that of the API and terminology of Hilltop. If you want more details about the internals of Hilltop, please look at the "Scripting.doc" file in the Hilltop installation folder if the doc is available to you.

2.1 Hilltop class

To work with the Hilltop class, first import the class and assign the **base_url** and **hts**. All data in Hilltop are stored in hts files. Consequently, one Regional Council may have hts files for different datasets.

```
from hilltoppy import Hilltop, utils

base_url = 'http://hilltop.gw.govt.nz/'
hts = 'data.hts'
```

The next step is to initialise the **Hilltop class**. This checks to see if the Hilltop server exists and that data can be retrieved from it. It will also throw an error if there are no sites available.

```
In [1]: ht = Hilltop(base_url, hts)
```

The Hilltop class uses the requests python package for sending and recieving data. You can pass any keyword args when initialising the Hilltop class to the requests.get function. For example, there are a couple Regional Councils that have issues with their SSL certificates. To make the Hilltop class work in this situation, you'll need to pass the verify=False parameter to the Hilltop class. But only do this if you have to.

```
ht = Hilltop(base_url, hts, verify=False)
```

The top level objects in Hilltop are **Sites**, which can be queried by calling the get_site_list method after the Hilltop class has been initialised. Calling it with only the base_url and hts will return all of the sites in an hts file. Adding the parameter location=True will return the Easting and Northing geographic coordinates (EPSG 2193), or location='LatLong' will return the Latitude and Longitude. There are other optional input parameters to get_site_list as well.

```
In [2]: sites_out1 = ht.get_site_list()

In [3]: sites_out1.head()
Out[3]:
```

(continues on next page)

(continued from previous page)

```
                              SiteName
0                               292611
1  Abbots Creek at D/S Donalds Crk Conf
2           Abbots Creek at Featherston
3            Abbots Creek at Lake Shore
4            Abbots Creek at SH2 Bridge

In [4]: sites_out2 = ht.get_site_list(location=True)

In [5]: sites_out2.head()
Out[5]:
                              SiteName    Easting   Northing
0                               292611        NaN        NaN
1  Abbots Creek at D/S Donalds Crk Conf  1793544.0  5441959.0
2           Abbots Creek at Featherston  1795280.0  5445581.0
3            Abbots Creek at Lake Shore  1792320.0  5441113.0
4            Abbots Creek at SH2 Bridge  1794031.0  5446253.0

In [6]: measurement = 'Total Phosphorus'

In [7]: sites_out3 = ht.get_site_list(location='LatLong',
   ...:                               measurement=measurement)
   ...:

In [8]: sites_out3.head()
Out[8]:
                            SiteName   Latitude   Longitude
0           Abbots Creek at Lake Shore -41.158519  175.292178
1           Abbots Creek at SH2 Bridge -41.111848  175.310929
2  Abbotts Creek at Longwood West Road -41.131190  175.323271
3   Akatarawa River at Hutt Confluence -41.089530  175.097661
4               Awhea River at Tora Rd -41.495662  175.511955
```

Using the get_site_info method on one or more sites will allow you to get a lot more site data than what's available via the get_site_list method.

```
In [9]: site = 'Akatarawa River at Hutt Confluence'

In [10]: site_data = ht.get_site_info(site)

In [11]: site_data
Out[11]:
                            SiteName      Agency  ... SecondSynonym SiteID
0  Akatarawa River at Hutt Confluence  Wellington  ...          RS25     49

[1 rows x 30 columns]
```

A Hilltop **Collection** groups one or more sites together. You can access all of the collections and associated sites via the get_collection_list method. Note that not all hts files (and organisations) have collections.

```
In [12]: collection1 = ht.get_collection_list()
```

(continues on next page)

(continued from previous page)

```
In [13]: collection1.head()
Out[13]:
        SiteName                      MeasurementName CollectionName
0  Birch Lane AQ              Air Quality PM10 (FH62)    Air Quality
1  Birch Lane AQ             Air Quality PM2.5 (FH62)    Air Quality
2  Birch Lane AQ   Air Quality PM10 (Hourly Average)    Air Quality
3  Birch Lane AQ    Air Quality PM10 (Daily Average)    Air Quality
4  Birch Lane AQ  Air Quality PM10 (Monthly Average)    Air Quality

In [14]: collection = 'WQ / Rivers and Streams'

In [15]: sites_out4 = ht.get_site_list(collection=collection)

In [16]: sites_out4.head()
Out[16]:
                                SiteName
0                  Awhea River at Tora Rd
1                Beef Creek at headwaters
2  Coles Creek tributary at Lagoon Hill Rd
3            Horokiri Stream at Snodgrass
4       Huangarua River at Ponatahi Bridge
```

As you can see, you can also pass a collection name to the get_site_list method to only get the sites in that collection.

The next step is to determine what types of **Measurements** are associated with the sites. This is where we call the get_measurement_list method to see all of the measurement names associated with one or more sites.

```
In [17]: site_meas = ht.get_measurement_list(site)

In [18]: site_meas.head()
Out[18]:
                            SiteName            MeasurementName  ...  \
0  Akatarawa River at Hutt Confluence               Derived Flow  ...
1  Akatarawa River at Hutt Confluence            Daily mean flow  ...
2  Akatarawa River at Hutt Confluence      Daily mean flow (4am)  ...
3  Akatarawa River at Hutt Confluence   Daily mean flow (Moving)  ...
4  Akatarawa River at Hutt Confluence  Derived Flow (24 Mov Avg)  ...

                    To  Divisor
0 2023-10-27 08:40:00      NaN
1 2023-10-27 08:40:00      NaN
2 2023-10-27 08:40:00      NaN
3 2023-10-27 08:40:00      NaN
4 2023-10-27 08:40:00      NaN

[5 rows x 15 columns]
```

There are a lot of data associated with Site/Measurement combos. These include Units, Precision, From, and To.

If all you want to know is what measurements exist in the hts file (regardless of the sites associated with them), there's a method for that! It does take some time for the Hilltop server to process this request though (and the Hilltop server might fail if the hts file is too big).

```
meas = ht.get_measurement_names()
```

Once you know the Site Name and Measurement Name you want time series data for, then you make a request via the get_data method. The get_data method has a variety of input parameters. Check the docstrings or package references for more details.

```
In [19]: measurement = 'Total Phosphorus'

In [20]: from_date = '2012-01-22 10:50'

In [21]: to_date = '2018-04-13 14:05'

In [22]: tsdata = ht.get_data(site, measurement, from_date=from_date,
   ....:                      to_date=to_date)
   ....:

In [23]: tsdata.head()
Out[23]:
                           SiteName   MeasurementName  ... Result Value  \
0  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
1  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
2  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
3  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
4  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN

   Standard Uncertainty
0                   NaN
1                   NaN
2                   NaN
3                   NaN
4                   NaN

[5 rows x 15 columns]
```

In addition to the time series value associated with the site and measurement, all other auxilliary data associated with the SiteName, MeasurementName, and Time will be returned. These auxilliary data can vary quite a bit and might not be consistant from one Regional Council to another.

If you run into an issue with your Hilltop server, you can debug via the browser by using the build_url function.

```
In [24]: url = utils.build_url(base_url, hts, 'MeasurementList', site)

In [25]: print(url)
http://hilltop.gw.govt.nz/data.hts?Service=Hilltop&Request=MeasurementList&Site=Akatarawa
↪%20River%20at%20Hutt%20Confluence
```

2.2 Legacy modules

Note: This section is only for achiving the legacy modules. Users should not normally use these. Please use the new Hilltop class described above.

2.2.1 Web service

The web service calls are simpler and more straightforward than the other two options. No extra setup is needed other than already having a Hilltop server to query. See the doc called "server.doc" for more details about the web service calls.

Data access

The function names are based on the associated Hilltop function names from the COM module. There is also an additional function specific to water quality samples. Below are an actual working examples!

Import the module and set the appropriate parameters.

```
from hilltoppy import web_service as ws

base_url = 'http://hilltop.gw.govt.nz/'
hts = 'data.hts'
site = 'Akatarawa River at Hutt Confluence'
collection = 'WQ / Rivers and Streams'
measurement = 'Total Phosphorus'
from_date = '2012-01-22 10:50'
to_date = '2018-04-13 14:05'
```

All data in Hilltop are stored in hts files. The top level objects in Hilltop are Sites, which can be queried by calling the site_list function. Calling it with only the base_url and hts will return all of the sites in an hts file. Adding the parameter location=True will return the Easting and Northing geographic coordinates (EPSG 2193), or location='LatLong' will return the Latitude and Longitude. There are other optional input parameters to site_list as well.

```
In [26]: sites_out1 = ws.site_list(base_url, hts)

In [27]: sites_out1.head()
Out[27]:
                             SiteName
0                              292611
1  Abbots Creek at D/S Donalds Crk Conf
2           Abbots Creek at Featherston
3            Abbots Creek at Lake Shore
4            Abbots Creek at SH2 Bridge

In [28]: sites_out2 = ws.site_list(base_url, hts, location=True)

In [29]: sites_out2.head()
Out[29]:
                             SiteName    Easting   Northing
0                              292611        NaN        NaN
```

(continues on next page)

(continued from previous page)

```
1  Abbots Creek at D/S Donalds Crk Conf  1793544.0  5441959.0
2           Abbots Creek at Featherston  1795280.0  5445581.0
3            Abbots Creek at Lake Shore  1792320.0  5441113.0
4            Abbots Creek at SH2 Bridge  1794031.0  5446253.0

In [30]: sites_out3 = ws.site_list(base_url, hts, location='LatLong',
   ....:                           measurement=measurement)
   ....:

In [31]: sites_out3.head()
Out[31]:
                            SiteName   Latitude   Longitude
0           Abbots Creek at Lake Shore -41.158519  175.292178
1           Abbots Creek at SH2 Bridge -41.111848  175.310929
2  Abbotts Creek at Longwood West Road -41.131190  175.323271
3   Akatarawa River at Hutt Confluence -41.089530  175.097661
4               Awhea River at Tora Rd -41.495662  175.511955
```

A Collection groups one or many Sites together and has its own function to return a dataframe of all the Sites and associated Collections. Note that not all hts files (and organisations) have collections.

```
In [32]: collection1 = ws.collection_list(base_url, hts)

In [33]: collection1.head()
Out[33]:
        SiteName                     MeasurementName CollectionName
0  Birch Lane AQ             Air Quality PM10 (FH62)    Air Quality
1  Birch Lane AQ            Air Quality PM2.5 (FH62)    Air Quality
2  Birch Lane AQ   Air Quality PM10 (Hourly Average)    Air Quality
3  Birch Lane AQ    Air Quality PM10 (Daily Average)    Air Quality
4  Birch Lane AQ  Air Quality PM10 (Monthly Average)    Air Quality

In [34]: sites_out4 = ws.site_list(base_url, hts, collection=collection)

In [35]: sites_out4.head()
Out[35]:
                               SiteName
0                  Awhea River at Tora Rd
1                 Beef Creek at headwaters
2  Coles Creek tributary at Lagoon Hill Rd
3             Horokiri Stream at Snodgrass
4       Huangarua River at Ponatahi Bridge
```

The next step is to determine what types of Measurements are associated with the Sites. In Hilltop, a Measurement is also associated to a Data Source. Conceptually, the Data Source represents the actual observation or measurement from the source, while the Measurement is a value derived from the Data Source. In many cases, the Measurement Name and the Data Source Name are the same, but there are instances where there are multiple Measurements per Data Source. For example, a Data Source Name of "Water Level" (which normally represents a surface water level) may have a Measurement Name of both Water Level and Flow (since flow can be derived from water level). Hilltop also has the concept of Virtual Measurements. Virtual Measurements do not have data directly stored in the hts files. Rather, Hilltop simply stores the equation to convert an existing Measurement (that does contain data) into a Virtual Measurement when the user requests the data. This reduces data storage with a very minor overhead computational cost.

In Hilltop, you must make a measurement_list function request to get all of the Data Sources and the associated Measurements.

```
In [36]: meas_df = ws.measurement_list(base_url, hts, site)

In [37]: meas_df.head()
Out[37]:
                            SiteName            MeasurementName  ...  \
0  Akatarawa River at Hutt Confluence             Derived Flow  ...
1  Akatarawa River at Hutt Confluence           Daily mean flow  ...
2  Akatarawa River at Hutt Confluence     Daily mean flow (4am)  ...
3  Akatarawa River at Hutt Confluence  Daily mean flow (Moving)  ...
4  Akatarawa River at Hutt Confluence  Derived Flow (24 Mov Avg)  ...

                   To  Divisor
0 2023-10-27 08:40:00      NaN
1 2023-10-27 08:40:00      NaN
2 2023-10-27 08:40:00      NaN
3 2023-10-27 08:40:00      NaN
4 2023-10-27 08:40:00      NaN

[5 rows x 15 columns]
```

Once you know the Site Name and Measurement Name you want time series data for, then you make a request via the get_data function. The get_data function has a variety of parameters. Check the doc strings or package references for more details.

```
In [38]: tsdata = ws.get_data(base_url, hts, site, measurement, from_date=from_date,
   ....:                       to_date=to_date)
   ....:

In [39]: tsdata.head()
Out[39]:
                            SiteName   MeasurementName  ... Result Value  \
0  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
1  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
2  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
3  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN
4  Akatarawa River at Hutt Confluence  Total Phosphorus  ...          NaN

   Standard Uncertainty
0                   NaN
1                   NaN
2                   NaN
3                   NaN
4                   NaN

[5 rows x 15 columns]
```

If you run into an issue with your Hilltop server, you can debug via the browser by using the build_url function.

```
In [40]: url = ws.build_url(base_url, hts, 'MeasurementList', site)

In [41]: print(url)
```

(continues on next page)

(continued from previous page)

```
http://hilltop.gw.govt.nz/data.hts?Service=Hilltop&Request=MeasurementList&Site=Akatarawa
↪%20River%20at%20Hutt%20Confluence
```

2.2.2 COM module

The following documentation describes how to set up and use the COM module functions. The COM module is no longer maintained!

Install pywin32

pywin32 does not come installed by default. Install it like any other python package before continuing.

```
conda install pywin32
```

Register Hydrolib

Hilltop Manager needs to be added into the Windows registry. This can be done for either the 32bit or the 64bit versions of Hilltop Manager, but if you have the choice pick the 64bit version in case you need to handle very large datasets. Find either version of Hilltop Manager, and open the program (called Manager.exe) as administrator. Load in an hts file (this allows you to access the configuration menus). Go to the tab called 'Configure' then go to 'installation'. It will ask you if you want Hilltop registered, and of course say yes.

Run makepy_hilltop

The COM utility must be built for hilltop to access it's functions. This is all wrapped in a single function. Once Hydrolib is properly registered, run makepy_hilltop without any parameters and you should be ready to use the COM functions.

```
from hilltoppy import com

com.makepy_hilltop()
```

Data access

The function names are based on the associated Hilltop function names. Since functionally, accessing quantity data is quite different (from the COM) as compared to the quality data, there are two functions accessing the time series data.

```
from hilltoppy import com

hts = r'\\path\to\file.hts'
sites = ['site1', 'site2']
mtypes = ['Total Suspended Solids']

meas_df = com.measurement_list(hts, sites)

tsdata = com.get_data_quality(hts, sites, mtypes)
print(tsdata)
```

2.2.3 Native Python module

The following documentation describes how to set up and use the module functions built upon the native python module. The Native Hilltop Python module is no longer maintained!

Python path to Hilltop.pyd

First, make sure that the Hilltop.pyd exists in either the root directory of the Hilltop directory or in the x64 directory (depending on your python installation). Open manager.exe, go to configure, and click on Python. It simply adds the Python path to the windows environment variables so that Python knows where to load the Hilltop.pyd from. This can also be modified from within Spyder or the sys module.

Data access

The function names are similar to the COM module except that one function covers both quantity and quality data.

```
from hilltoppy import hilltop

hts = r'\\path\to\file.hts'
sites = ['site1', 'site2']
mtypes = ['Total Suspended Solids']

sites_out = hilltop.site_list(hts)

meas_df = hilltop.measurement_list(hts, sites)

tsdata = hilltop.get_data(hts, sites, mtypes)
print(tsdata)
```

CHAPTER

THREE

PACKAGE REFERENCES

3.1 Hilltop class

class hilltoppy.**Hilltop**(*base_url: str*, *hts: str*, *timeout: int = 60*, ***kwargs*)

get_collection_list()

CollectionList request method. Returns a dataframe of collection and site names associated with the hts file.

Return type
DataFrame

get_data(*sites: str | List[str]*, *measurements: str | List[str]*, *from_date: str | None = None*, *to_date: str | None = None*, *agg_method: str | None = None*, *agg_interval: str | None = None*, *alignment: str = '00:00'*, *quality_codes: bool = False*, *apply_precision: bool = False*, *tstype: str | None = None*)

Method to query a Hilltop web server for time series data associated with a Site and Measurement.

Parameters

- **sites** (*str or list of str*) – The site(s) to get the results. You can pass a single site as a string, or a list of sites.
- **measurements** (*str or list of str*) – The measurement(s) to get the results. If multiple sites and measurements are passed, all combinations must exist in Hilltop.
- **from_date** (*str or None*) – The start date in the format 2001-01-01. None will put it to the beginning of the time series.
- **to_date** (*str or None*) – The end date in the format 2001-01-01. None will put it to the end of the time series.
- **agg_method** (*str or None*) – The aggregation method to resample the data. e.g. Average, Total, Moving Average, Extrema.
- **agg_interval** (*str or None*) – The aggregation interval for the agg_method. e.g. '1 day', '1 week', '1 month'.
- **alignment** (*str or None*) – The start time alignment when agg_method is not None.
- **quality_codes** (*bool*) – Should the quality codes get returned?
- **apply_precision** (*bool*) – Should the precision according to Hilltop be applied to the data? Only use True if you're confident that Hilltop stores the correct precision, because it is not always correct.
- **tstype** (*str or None*) – The time series type; one of Standard, Check, or Quality.

Return type
DataFrame

`get_measurement_list`(*sites: str | List[str] | None = None*, *measurement: str | None = None*)

Method to query a Hilltop server for the measurement summary of a site or sites.

Parameters

- **sites** (*str, list of str, or None*) – The site(s) to get the measurements. You can pass a single site as a string, a list of sites, or None to get the measurements for all available sites in the hts file.
- **measurement** (*str or None*) – The measurement name to filter the sites by.

Return type
DataFrame

`get_measurement_names`(*detailed=False*)

Method to get all of the available Measurement Names in the hts. When detailed=False, then the request is relatively fast but only returns the names. When detailed=True, the method runs through as many sites as necessary to get additional data about the Measurements.

Parameters
detailed (*bool*) – If True, the method runs through as many sites as necessary to get additional data about the Measurements. It may take several minutes to run this query.

Return type
DataFrame

`get_site_info`(*sites: str | List[str] | None = None*)

SiteInfo request function. Returns all of the site data for a specific site. The Hilltop sites table has tons of fields, so you never know what you're going to get.

Parameters
sites (*str, list of str, or None*) – The site(s) to get the site info. You can pass a single site as a string, a list of sites, or None to get the site info for all available sites in the hts file.

Return type
DataFrame

`get_site_list`(*location: str | bool | None = None*, *measurement: str | None = None*, *collection: str | None = None*, *site_parameters: List[str] | None = None*)

SiteList request function. Returns a list of sites associated with the hts file.

Parameters

- **location** (*str, bool, or None*) – Should the location be returned? Only applies to the SiteList request. 'Yes' returns the Easting and Northing, while 'LatLong' returns NZGD2000 lat lon coordinates.
- **measurement** (*str or None*) – The measurement name.
- **collection** (*str or None*) – Get site list via a collection.
- **site_parameters** (*list or None*) – A list of the site parameters to be returned with the SiteList request. Make a call to site_info to find all of the possible options.

Return type
DataFrame

3.2 Legacy modules

3.2.1 COM module

hilltoppy.com.**makepy_hilltop**(*hlib='Hilltop Data Access'*)

Function to generate the Hilltop COM module.

Parameters
hlib (*str*) – The name of the COM library.

Return type
None

hilltoppy.com.**measurement_list**(*hts*, *sites=None*, *mtypes=None*, *rem_wq_sample=True*)

Function to read the site names, measurement types, and units of a Hilltop hts file. Returns a DataFrame.

Parameters

- **hts** (*str*) – Path to the hts file.
- **sites** (*list or None*) – A list of site names within the hts file.
- **mtypes** (*list or None*) – A list of measurement types that should be returned.
- **rem_wq_sample** (*bool*) – In Hilltop 'WQ Sample' is a measurement type placemarker for the additional sample data. It doesn't generally apply when querying for the combo of sites/measurement types. True removes this instance from the returned DataFrame.

Return type
DataFrame

hilltoppy.com.**get_data_quantity**(*hts*, *sites=None*, *mtypes=None*, *start=None*, *end=None*, *agg_period=None*, *agg_n=1*, *fun=None*, *output_site_data=False*, *exclude_mtype=None*, *sites_df=None*)

Function to read water quantity data from an hts file.

Parameters

- **hts** (*str*) – Path to the hts file.
- **sites** (*list*) – A list of site names within the hts file.
- **mtypes** (*list*) – A list of measurement types that should be returned.
- **start** (*str*) – The start date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **end** (*str*) – The end date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **agg_period** (*str*) – The resample period (e.g. 'day', 'month').
- **agg_n** (*int*) – The number of periods (e.g. 1 for 1 day).
- **fun** (*str*) – The resampling function.
- **output_site_data** (*bool*) – Should the sites data be output?
- **sites_df** (*DataFrame*) – The DataFrame return from the rd_hilltop_sites function. If this is passed than rd_hilltop_sites is not run.

Return type
DataFrame

hilltoppy.com.**get_data_quality**(*hts*, *sites=None*, *mtypes=None*, *start=None*, *end=None*, *dtl_method=None*, *output_site_data=False*, *mtype_params=None*, *sample_params=None*, *sites_df=None*)

Function to read water quality data from an hts file.

Parameters

- **hts** (*str*) – Path to the hts file.
- **sites** (*list*) – A list of site names within the hts file.
- **mtypes** (*list*) – A list of measurement types that should be returned.
- **start** (*str*) – The start date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **end** (*str*) – The end date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **dtl_method** (*None, 'standard', 'trend'*) – The method to use to convert values under a detection limit to numeric. None does no conversion. 'standard' takes half of the detection limit. 'trend' is meant as an output for trend analysis with includes an additional column dtl_ratio referring to the ratio of values under the detection limit.
- **output_site_data** (*bool*) – Should the site data be output?
- **sites_df** (*DataFrame*) – The DataFrame return from the rd_hilltop_sites function. If this is passed than rd_hilltop_sites is not run.

Return type
DataFrame

3.2.2 Native Hilltop Python module

hilltoppy.hilltop.**site_list**(*hts*)

Function to return a list of sites from an hts file. Exists for consistancy.

Parameters
hts (*str*) – Path to hts file.

Return type
list

hilltoppy.hilltop.**measurement_list**(*hts*, *sites=None*)

Function to read all of the sites in an hts file and the associated site info.

Parameters

- **hts** (*str*) – Path to hts file.
- **sites** (*list*) – A list of site names to return.

Return type
DataFrame

hilltoppy.hilltop.**get_data**(*hts*, *sites=None*, *mtypes=None*, *from_date=None*, *to_date=None*, *agg_method='Average'*, *agg_n=1*, *agg_period='day'*, *output_missing_sites=False*, *site_info=None*)

Function to read time series from an hts file.

Parameters

- **hts** (*str*) – Path to the hts file.

- **sites** (*list*) – A list of site names within the hts file.
- **mtypes** (*list*) – A list of measurement types that should be returned.
- **from_date** (*str*) – The start date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **to_date** (*str*) – The end date to retreive from the data in ISO format (e.g. '2011-11-30 00:00').
- **agg_method** (*str*) – Options are '', 'Interpolate', 'Average', 'Total', 'Moving Average', and 'EP'.
- **agg_period** (*str*) – The resample period (e.g. 'day', 'month').
- **agg_n** (*int*) – The number of periods (e.g. 1 for 1 day).
- **output_site_data** (*bool*) – Should the sites data be output?
- **sites_info** (*DataFrame*) – The DataFrame return from the get_sites_mtypes function. If this is passed than get_sites_mtypes is not run.

Return type
DataFrame

3.2.3 Web service

hilltoppy.web_service.**build_url**(*base_url: str*, *hts: str*, *request: str*, *site: str | None = None*, *measurement: str | None = None*, *collection: str | None = None*, *from_date: str | None = None*, *to_date: str | None = None*, *location: str | bool | None = None*, *site_parameters: List[str] | None = None*, *agg_method: str | None = None*, *agg_interval: str | None = None*, *alignment: str | None = None*, *quality_codes: bool = False*, *tstype: str | None = None*, *response_format: str | None = None*, *units: bool | None = None*)

Function to generate the Hilltop url for the web service.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension. Even if the file to be accessed is a dsn file, it must still have an hts extension for the web service.
- **request** (*str*) – The function request.
- **site** (*str or None*) – The site to be extracted.
- **measurement** (*str or None*) – The measurement type name.
- **collection** (*str or None*) – The collection name.
- **from_date** (*str or None*) – The start date in the format 2001-01-01. None will put it to the beginning of the time series.
- **to_date** (*str or None*) – The end date in the format 2001-01-01. None will put it to the end of the time series.
- **location** (*str or bool*) – Should the location be returned? Only applies to the SiteList request. True returns the Easting and Northing, while 'LatLong' returns NZGD2000 lat lon coordinates.
- **site_parameters** (*list of str*) – A list of the site parameters to be returned with the SiteList request.

- **agg_method** (*str*) – The aggregation method to resample the data. e.g. Average, Total, Moving Average, Extrema.
- **agg_interval** (*str*) – The aggregation interval for the agg_method. e.g. '1 day', '1 week', '1 month'.
- **alignment** (*str*) – The time alignment in the form '00:00'.
- **quality_codes** (*bool*) – Should the quality codes get returned from the GetData function.
- **tstype** (*str*) – The timeseries type, one of Standard, Check or Quality
- **response_format** (*str*) – The Hilltop response structure. Options are None, Native, or WML2. Read the Hilltop server docs for more info.

Returns
URL string for the Hilltop web server.

Return type
str

hilltoppy.web_service.**site_list**(*base_url*, *hts*, *location=None*, *measurement=None*, *collection=None*, *site_parameters=None*, *timeout=60*, ***kwargs*)

SiteList request function. Returns a list of sites associated with the hts file.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension.
- **location** (*str or bool*) – Should the location be returned? Only applies to the SiteList request. 'Yes' returns the Easting and Northing, while 'LatLong' returns NZGD2000 lat lon coordinates.
- **collection** (*str*) – Get site list via a collection.
- **site_parameters** (*list*) – A list of the site parameters to be returned with the SiteList request. Make a call to site_info to find all of the possible options.
- **timeout** (*int*) – The http request timeout in seconds.
- ****kwargs** – Optional keyword arguments passed to requests.

Return type
DataFrame

hilltoppy.web_service.**site_info**(*base_url*, *hts*, *site*, *timeout=60*, ***kwargs*)

SiteInfo request function. Returns all of the site data for a specific site. The Hilltop sites table has tons of fields, so you never know what you're going to get.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension.
- **site** (*str or None*) – The site to be extracted.
- **timeout** (*int*) – The http request timeout in seconds.
- ****kwargs** – Optional keyword arguments passed to requests.

Return type
DataFrame

hilltoppy.web_service.**collection_list**(*base_url*, *hts*, *timeout=60*, ***kwargs*)

CollectionList request function. Returns a frame of collection and site names associated with the hts file.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension.
- **timeout** (*int*) – The http request timeout in seconds.
- ****kwargs** – Optional keyword arguments passed to requests.

Return type
DataFrame

hilltoppy.web_service.**measurement_list**(*base_url*, *hts*, *site*, *measurement=None*, *timeout=60*, ***kwargs*)

Function to query a Hilltop server for the measurement summary of a site.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension. Even if the file to be accessed is a dsn file, it must still have an hts extension for the web service.
- **site** (*str or None*) – The site to be extracted.
- **measurement** (*str or None*) – The measurement type name.
- **timeout** (*int*) – The http request timeout in seconds.
- ****kwargs** – Optional keyword arguments passed to requests.

Return type
DataFrame

hilltoppy.web_service.**get_data**(*base_url*, *hts*, *site*, *measurement*, *from_date=None*, *to_date=None*, *agg_method=None*, *agg_interval=None*, *alignment='00:00'*, *quality_codes=False*, *apply_precision=False*, *tstype=None*, *timeout=60*, ***kwargs*)

Function to query a Hilltop web server for time series data associated with a Site and Measurement.

Parameters

- **base_url** (*str*) – root Hilltop url str.
- **hts** (*str*) – hts file name including the .hts extension. Even if the file to be accessed is a dsn file, it must still have an hts extension for the web service.
- **request** (*str*) – The function request.
- **site** (*str or None*) – The site to be extracted.
- **measurement** (*str or None*) – The measurement type name.
- **from_date** (*str or None*) – The start date in the format 2001-01-01. None will put it to the beginning of the time series.
- **to_date** (*str or None*) – The end date in the format 2001-01-01. None will put it to the end of the time series.
- **agg_method** (*str*) – The aggregation method to resample the data. e.g. Average, Total, Moving Average, Extrema.

- **agg_interval** (*str*) – The aggregation interval for the agg_method. e.g. '1 day', '1 week', '1 month'.
- **alignment** (*str*) – The start time alignment when agg_method is not None.
- **quality_codes** (*bool*) – Should the quality codes get returned?
- **apply_precision** (*bool*) – Should the precision according to Hilltop be applied to the data? Only use True if you're confident that Hilltop stores the correct precision, because it is not always correct.
- **tstype** (*str or None*) – The time series type; one of Standard, Check, or Quality.
- **timeout** (*int*) – The http request timeout in seconds.
- ****kwargs** – Optional keyword arguments passed to requests.

Return type
DataFrame

3.3 API Pages

CHAPTER

FOUR

LICENSE AND TERMS OF USAGE

This package is licensed under the terms of the Apache License Version 2.0 and can be found on the GitHub project page.

INDEX