
kismetdb Documentation

Release 2019.05.02

Ash Wilson

May 24, 2019

Contents

1	Kismet database wrapper	1
1.1	Quickstart	1
1.2	Included scripts	1
1.3	Testing	2
2	Table of Contents	3
2.1	Tables	3
2.2	Extras	9
2.3	Testing	11
2.4	Updating and Extending	11
2.5	Changelog	13
3	Indices and tables	15

1.1 Quickstart

Install from PyPI with `pip install kismetdb`

Install from source with `pip install .`

In the Python interpreter:

```
import json
import kismetdb
kismet_log_file = "kismet/database.here"
alerts = kismetdb.Alerts(kismet_log_file)

# Get alert metadata
all_alerts_meta = alerts.get_meta()
for alert in all_alerts_meta:
    print(alert["header"])

# Get payload from all alerts
all_alerts = alerts.get_all()
for alert in all_alerts:
    print(json.loads(alert["json"])["kismet.alert.text"])
```

1.2 Included scripts

Alongside the Python library, several commands are installed:

- `kismet_log_devices_to_json`
- `kismet_log_to_csv`

- `kismet_log_to_kml`
- `kismet_log_to_pcap`
- `kismet_log_devices_to_filebeat_json`

Following any of the prior commands with `--help` will provide details on usage.

1.3 Testing

In order to test, you must place a kismet sqlite log file at `tests/assets/testdata.kismet_4` and `tests/assets/testdata.kismet_5`, which are Kismet version 4 and 5 databases, respectively.

Testing happens in a Docker build process:

Testing for Python 2.7:

```
docker build .
```

Testing for Python 3.6:

```
docker build --build-arg PY_VER=3.6 .
```

Testing for Python 3.7:

```
docker build --build-arg PY_VER=3.7 .
```

2.1 Tables

This wrapper presents tables as Python objects.

2.1.1 Alerts

class `kismetdb.Alerts` (*file_location*)

This object covers alerts stored in the Kismet DB.

The `Keyword Arguments` section below applies only to methods which support them (as noted below), not to object instantiation.

Parameters `file_location` (*str*) – Path to Kismet log file.

Keyword Arguments

- **ts_sec_gt** (*str, datetime, or (secs, u_secs)*) – Timestamp for starting query.
- **phyname** (*str, list*) – Restrict results to this PHY.
- **devmac** (*str, list*) – Restrict results to this MAC address.
- **header** (*str, list*) – Restrict results to alerts of this type.

get_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (***kwargs*)

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (***kwargs*)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.1.2 DataSources

class kismetdb.DataSources (*file_location*)

This object covers data sources stored in the Kismet DB.

The Keyword Arguments section below applies only to methods which support them (as noted below), not to object instantiation.

Parameters *file_location* (*str*) – Path to Kismet log file.

Keyword Arguments

- **uuid** (*str*, *list*) – UUID of data source.
- **typestring** (*str*, *list*) – Type of data source.
- **definition** (*str*, *list*) – Data source definition.
- **name** (*str*, *list*) – Name of data source.
- **interface** (*str*, *list*) – Interface associated with data source.

get_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (***kwargs*)

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (***kwargs*)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.1.3 Devices

class kismetdb.Devices (*file_location*)

This object covers devices tracked in the Kismet DB.

Unlike other abstractions which contain the object detail under the *json* key, this abstraction contains the details under the key named *device*. The **Keyword Arguments** section below applies only to methods which support them (as noted below), not to object instantiation.

Parameters *file_location* (*str*) – Path to Kismet log file.

Keyword Arguments

- **first_time_lt** (*str*, *datetime.datetime*) – Match devices where the first observation timestamp is before this time.
- **first_time_gt** (*str*, *datetime.datetime*) – Match devices where the first observation timestamp is after this time.
- **last_time_lt** (*str*, *datetime.datetime*) – Match devices where the most recent observation timestamp is before this time.
- **last_time_gt** (*str*, *datetime.datetime*) – Match devices where the most recent observation timestamp is after this time.
- **devkey** (*str*, *list*) – Exact match for this devkey.
- **phyname** (*str*, *list*) – Exact match for this phyname.
- **devmac** (*str*, *list*) – Exact match for this device MAC.
- **type** (*str*, *list*) – Exact match for this device type.
- **strongest_signal_gt** (*str*, *int*) – Match devices where the strongest signal is greater than the integer representation of this string.
- **strongest_signal_lt** (*str*, *int*) – Match devices where the strongest signal is less than the integer representation of this string.
- **bytes_data_gt** (*str*, *int*) – Match devices where we've seen at least this many bytes of data (converted to int).
- **bytes_data_lt** (*str*, *int*) – Match devices where we've seen at most this many bytes of data (converted to int).

get_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (***kwargs*)

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (***kwargs*)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.1.4 Kismet

class kismetdb.**Kismet** (*filepath*)

This object extracts kismet server info from the first SYSTEM snapshot in the database. All values reference the Kismet server which generated this log.

Parameters **file_location** (*str*) – Path to Kismet log file.

kismet_version

Kismet version

Type str

kismet_git

Kismet git commit string

Type str

kismet_uuid

UUID of server

Type str

kismet_name

User-supplied name of server

Type str

kismet_location
User-supplied server location
Type str

kismet_description
User-supplied server description
Type str

kismet_user
Username server was running under
Type str

2.1.5 Packets

class kismetdb.Packets (*file_location*)

This object covers packets stored in the Kismet DB.

The actual packet is stored in the *packet* field of the dictionary returned for every row. This can be a very expensive abstraction to use if you don't employ some sort of filtering on your query. Consider using the *Packets.get_meta()* method to retrieve only the metadata (not the actual packet capture), which will preserve performance. The Keyword Arguments section below applies only to methods which support them (as noted below), not to object instantiation.

Parameters *file_location* (*str*) – Path to Kismet log file.

Keyword Arguments

- **ts_sec_lt** (*str*, *datetime.datetime*) – Match packets where the timestamp is before this.
- **ts_sec_gt** (*str*, *datetime.datetime*) – Match packets where the timestamp is after this.
- **phyname** (*str* or *list*) – Exact match against PHY name.
- **sourcemac** (*str* or *list*) – Exact match against source MAC address.
- **destmac** (*str* or *list*) – Exact match against destination MAC address.
- **transmac** (*str* or *list*) – Exact match against trans mac.
- **devkey** (*str* or *list*) – Exact match against devkey.
- **datasource** (*str* or *list*) – Exact match against datasource.
- **min_signal** (*str* or *int*) – Minimum signal.
- **dlt_gt** (*str* or *int*) – Minimum DLT.

get_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (***kwargs*)

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (***kwargs*)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.1.6 Snapshots

class kismetdb.**Snapshots** (*file_location*)

This object covers snapshots stored in the Kismet DB.

The `Keyword Arguments` section below applies only to methods which support them (as noted below), not to object instantiation.

Parameters **file_location** (*str*) – Path to Kismet log file.

Keyword Arguments

- **ts_sec_gt** (*str, datetime, or (secs, u_secs)*) – Timestamp for starting query.
- **ts_sec_lt** (*str, datetime, or (secs, usecs)*) – Timestamp for ending query.
- **lat_gt** (*str, float*) – Bounding minimum latitude
- **lat_lt** (*str, float*) – Bounding maximum latitude
- **lon_gt** (*str, float*) – Bounding minimum longitude
- **lon_lt** (*str, float*) – Bounding maximum longitude
- **snaptype** (*str*) – Snapshot type

get_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (***kwargs*)

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (**kwargs)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (**kwargs)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.2 Extras

Some pre-built scripts are included for common use cases.

2.2.1 kismet_log_devices_to_filebeat_json

Export from the `devices` table to stdout or append a json file.

```
usage: kismet_log_devices_to_filebeat_json [-h] --in INFILE [--out OUTFILE]
                                           [--start-time STARTTIME]
                                           [--min-signal MINSIGNAL]

optional arguments:
-h, --help                show this help message and exit
--in INFILE               Input (.kismet) file
--out OUTFILE             Output filename (optional) for appending. If unspecified,
                           each record will be printed to stdout, one record per line,
                           ideal for piping into filebeat.
--start-time STARTTIME    Only list devices seen after given time
--min-signal MINSIGNAL    Only list devices with a best signal higher than min-signal
```

2.2.2 kismet_log_devices_to_json

Export contents of `devices` table in Kismet DB to json file.

```
usage: kismet_log_devices_to_json [-h] [--in INFILE] [--out OUTFILE]
                                   [--start-time STARTTIME]
                                   [--min-signal MINSIGNAL]

optional arguments:
-h, --help                show this help message and exit
--in INFILE               Input (.kismet) file
--out OUTFILE             Output filename (optional). If omitted, logs multi-
                           line and indented (human-readable) to stdout.
--start-time STARTTIME    Only list devices seen after given time
--min-signal MINSIGNAL    Only list devices with a best signal higher than min-signal
```

2.2.3 kismet_log_to_csv

Export contents of various tables in Kismet DB to csv file.

```
usage: kismet_log_to_csv [-h] [--in INFILE] [--out OUTFILE] [--table SRCTABLE]

optional arguments:
  -h, --help            show this help message and exit
  --in INFILE           Input (.kismet) file
  --out OUTFILE         Output CSV filename
  --table SRCTABLE      Select the table to export. The ``packets``, ``datasources``,
                        and ``alerts`` tables are supported. Defaults to ``devices``
                        ↪table.
```

2.2.4 kismet_log_to_kml

Export contents of the devices table to KML.

```
usage: kismet_log_to_kml [-h] [--in INFILE] [--out OUTFILE]
                        [--start-time STARTTIME] [--min-signal MINSIGNAL]
                        [--strongest-point] [--title TITLE] [--ssid SSID]

optional arguments:
  -h, --help            show this help message and exit
  --in INFILE           Input (.kismet) file
  --out OUTFILE         Output filename (optional)
  --start-time STARTTIME Only list devices seen after given time
  --min-signal MINSIGNAL Only list devices with a best signal higher than min-signal
  --strongest-point      Plot points based on strongest signal
  --title TITLE         Title embedded in KML file
  --ssid SSID           Only plot networks which match the SSID (or SSID regex)
```

2.2.5 kismet_log_to_pcap

Export captures from the packets table to .pcap file.

```
usage: kismet_log_to_pcap [-h] [--in INFILE] [--out OUTFILE]
                        [--outtitle OUTTITLE] [--limit-packets LIMITPACKETS]
                        [--source-uuid UUID] [--start-time STARTTIME]
                        [--end-time ENDTIME] [--silent SILENT]
                        [--min-signal MINSIGNAL] [--device-key DEVICEKEY]

optional arguments:
  -h, --help            show this help message and exit
  --in INFILE           Input (.kismet) file
  --out OUTFILE         Output filename (when exporting all packets)
  --outtitle OUTTITLE   Output title (when limiting packets per file)
  --limit-packets LIMITPACKETS Generate multiple pcap files, limiting the number
                        of packets per file
  --source-uuid UUID     Limit packets to a specific data source (multiple
                        --source-uuid options will match multiple
                        ↪datasources)
  --start-time STARTTIME Only convert packets recorded after start-time
```

(continues on next page)

(continued from previous page)

<code>--end-time</code> ENDTIME	Only convert packets recorded before end-time
<code>--silent</code> SILENT	Silent operation (no status output)
<code>--min-signal</code> MINSIGNAL	Only convert packets with a signal greater than min-
↪signal	
<code>--device-key</code> DEVICEKEY	Only convert packets which are linked to the ↪
↪specified device	key (multiple <code>--device-key</code> options will match ↪
↪multiple devices)	

2.3 Testing

In order to test, you must place a kismet sqlite log file at `tests/assets/testdata.kismet_4` and `tests/assets/testdata.kismet_5`, which are version 4 and version 5 log files, respectively.

Testing happens in a Docker build process:

Testing for Python 2.7:

```
docker build .
```

Testing for Python 3.6:

```
docker build --build-arg PY_VER=3.6 .
```

Testing for Python 3.7:

```
docker build --build-arg PY_VER=3.7 .
```

2.4 Updating and Extending

Over time, we expect that the database schema will change. To make transitioning to a new schema easier, each object is defined with the expected database columns defined in a class variable named `column_names`. The bulk data field (which contains json or raw packet capture) is in a class variable named `bulk_data_field`. The `valid_kwargs` class variable is used in parsing keyword arguments for filtering in the SQL query. These items tie into functions that live in the Utility class, and are used for forming the SQL that's used to query the Kismet DB.

This tool follows calendar versioning, and new versions support DB schemas as far back as v4.

As the database schema changes, the changes required to support a new version of the db will be required on a per-object basis. The following object attributes are used to contain version-specific schema information:

- `field_defaults`: This is used to force a default value for fields that are not found in older-than-current versions of the Kismet DB.
- `converters_reference`: This allows us to specify a converter so that if the data type changes between schema versions, we can force the older DB type to match the current DB version's type.
- `column_reference`: This describes the expected columns for each supported version of the kismet DB

All objects representing tables inherit from the `BaseInterface` class:

```
class kismetdb.BaseInterface (file_location)
```

Initialize with a path to a valid Kismet log file.

Parameters `file_location` (*str*) – Path to Kismet log file.

bulk_data_field

Field containing bulk data (typically stored as a blob in the DB). This allows the *get_meta()* method to exclude information which may have a performance impact. This is especially true for the retrieval of packet captures.

Type str

column_reference

Top-level keys in this dictionary are version numbers, and are used to easily extend the schema for new versions. The `column_names` attribute is populated from this during instantiation.

Type dict

column_names

Name of columns expected to be in this object's table by this abstraction. Used for validation against columns in DB on instantiation.

Type list

table_name

Name of the table this abstraction represents.

Type str

valid_kwargs

This is a dictionary where the key is the name of a keyword argument and the value is a reference to the function which builds the SQL partial and replacement dictionary.

Type str

field_defaults

Statically set these column defaults by DB version.

Type dict

converters_reference

This provides a reference for converters to use on data coming from the DB on a version by version basis.

Type dict

full_query_column_names

Processed column names for full query of kismet DB. Created on instantiation.

Type list

meta_query_column_names

Processed column names for meta query of kismet DB. Created on instantiation.

Type list

get_all (kwargs)**

Get all objects represented by this class from Kismet DB.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

get_meta (kwargs)**

Get metadata columns from DB, excluding bulk data columns.

Keyword arguments are described above, near the beginning of the class documentation.

Returns List of each json object from all rows returned from query.

Return type list

yield_all (***kwargs*)

Get all objects represented by this class from Kismet DB.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Yields *dict* – Dict representing one row from query.

yield_meta (***kwargs*)

Yield metadata from DB, excluding bulk data columns.

Yields one row at a time. Keyword arguments are described above, near the beginning of the class documentation.

Returns Dict representing one row from query.

Return type dict

2.5 Changelog

2.5.1 v2019.05.02

- Make RST doc levels happy. [Mike Kershaw / Dragorn]
- Hopefully make docs happy. [Mike Kershaw / Dragorn]
- Add self to docs. [Mike Kershaw / Dragorn]
- Fix changelog. [Mike Kershaw / Dragorn]
- Fix RST? [Mike Kershaw / Dragorn]
- Docs. [Mike Kershaw / Dragorn]
- Ignore vim. [Mike Kershaw / Dragorn]
- Enable classes Bump version Add integer version. [Mike Kershaw / Dragorn]
- Add snapshots class Add kismet class for server info derived from snapshots. [Mike Kershaw / Dragorn]
- Add float comparators Add string LIKE comparators. [Mike Kershaw / Dragorn]
- Add defaults for db6. [Mike Kershaw / Dragorn]
- Add support for database version 6. [Mike Kershaw / Dragorn]
- Add license file now that it's a submodule. [Mike Kershaw / Dragorn]
- Minor commit to trigger mirror. [Mike Kershaw / Dragorn]

2.5.2 v5.1.0 (2019-02-16)

New

- Include version-specific converters. [Ash Wilson]

This allows us to, for instance, ensure that all GPS coordinates are returned as float-type values, across all database versions, no matter how they were originally stored in the database.

Closes #22

- Support v4 as well as v5 Kismet databases. [Ash Wilson]
Closes #19
- Add `kismet_log_devices_to_filebeat_json`. [Ash Wilson]
Closes #17

2.5.3 v5.0.0 (2019-02-12)

New

- Support v5 schema. [Ash Wilson]

2.5.4 v4.0.3 (2019-02-05)

Changes

- Updated docs, added simplekml requirement. [Ash Wilson]
Closes #8 Closes #7
- Adding docs to be built by Sphinx. [Ash Wilson]
- Scripts automatically install with Python package. [Ash Wilson]
Added generator function `yield_rows()` to all abstractions.
- Initial working commit. [Ash Wilson]
In order to run integration tests, you need a Kismet db at `tests/assets/testdata.kismet`.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

A

Alerts (*class in kismetdb*), 3

B

BaseInterface (*class in kismetdb*), 11

bulk_data_field (*kismetdb.BaseInterface attribute*), 11

C

column_names (*kismetdb.BaseInterface attribute*), 12

column_reference (*kismetdb.BaseInterface attribute*), 12

converters_reference (*kismetdb.BaseInterface attribute*), 12

D

DataSources (*class in kismetdb*), 4

Devices (*class in kismetdb*), 5

F

field_defaults (*kismetdb.BaseInterface attribute*), 12

full_query_column_names (*kismetdb.BaseInterface attribute*), 12

G

get_all() (*kismetdb.Alerts method*), 3

get_all() (*kismetdb.BaseInterface method*), 12

get_all() (*kismetdb.DataSources method*), 4

get_all() (*kismetdb.Devices method*), 5

get_all() (*kismetdb.Packets method*), 7

get_all() (*kismetdb.Snapshots method*), 8

get_meta() (*kismetdb.Alerts method*), 3

get_meta() (*kismetdb.BaseInterface method*), 12

get_meta() (*kismetdb.DataSources method*), 4

get_meta() (*kismetdb.Devices method*), 6

get_meta() (*kismetdb.Packets method*), 7

get_meta() (*kismetdb.Snapshots method*), 8

K

Kismet (*class in kismetdb*), 6

kismet_description (*kismetdb.Kismet attribute*), 7

kismet_git (*kismetdb.Kismet attribute*), 6

kismet_location (*kismetdb.Kismet attribute*), 6

kismet_name (*kismetdb.Kismet attribute*), 6

kismet_user (*kismetdb.Kismet attribute*), 7

kismet_uuid (*kismetdb.Kismet attribute*), 6

kismet_version (*kismetdb.Kismet attribute*), 6

M

meta_query_column_names (*kismetdb.BaseInterface attribute*), 12

P

Packets (*class in kismetdb*), 7

S

Snapshots (*class in kismetdb*), 8

T

table_name (*kismetdb.BaseInterface attribute*), 12

V

valid_kwargs (*kismetdb.BaseInterface attribute*), 12

Y

yield_all() (*kismetdb.Alerts method*), 4

yield_all() (*kismetdb.BaseInterface method*), 13

yield_all() (*kismetdb.DataSources method*), 4

yield_all() (*kismetdb.Devices method*), 6

yield_all() (*kismetdb.Packets method*), 8

yield_all() (*kismetdb.Snapshots method*), 9

yield_meta() (*kismetdb.Alerts method*), 4

yield_meta() (*kismetdb.BaseInterface method*), 13

yield_meta() (*kismetdb.DataSources method*), 5

yield_meta() (*kismetdb.Devices method*), 6

yield_meta() (*kismetdb.Packets method*), 8

yield_meta() (*kismetdb.Snapshots method*), 9