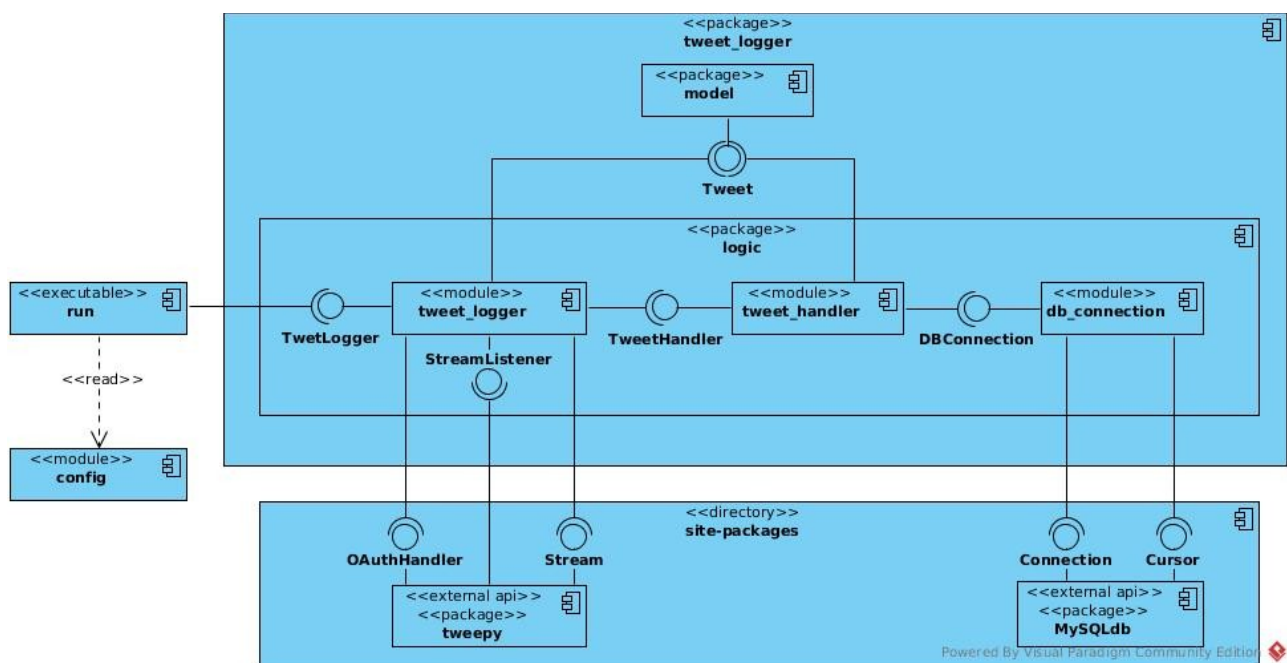


Glossario:

- Package:** Directory that contains an `__init__.py` file and any number of modules and sub-packages
- Module:** File with `.py` extension that declares any number of variables, functions and classes.
- Executable:** Module that is designed to be executed by the Python interpreter.

Components:

This diagram shows the interfaces implemented and required by each module. The **dmbs_handling** module is designed to offer multiple implementations of the DBConnection interface. Final users can set the desired implementation in the `config.py` module. Current and only implementation relies on MySQLdb API.



Interfaces:

Here follows the specification of the interfaces defined in the **tweet_logger** package. Other interfaces are documented in the relative API documentation.

class **TweetLogger**:

```
def __init__ (self, **config):
    """
    all config options are described in the config module. Most relevant are:
    credentials to access Twitter API, db backend to use, credentials to access db.
    -----
    Authenticate to Twitter API and connect with db
    """

def start (self, filter):
    """
    @filter: list of str
    -----
    stream from Twitter API all the tweets matching the keywords in filter in real time
    saving tweets data on db as they arrive.
    This method is blocking and never returning. The only way to stop is with KeyboardInterrupt!
    """
```

class **TweetHandler**:

```
def __init__ (self, **config):
    """
    all config options are described in the config module. most relevant are:
    db backend to use, credentials to access db, log file path.
    -----
    Connect with db
    """

def handle_tweet (self, tweet):
    """
    @tweet: Tweet
    -----
    Save tweet on db. Log any error to the log file specified in the init config.
    """

def handle_connection_error (self, errno):
    """
    @errno: int
    -----
    print errno to stderr and log it to the log file specified in the init config.
    """
```

class **DBConnection**:

 @staticmethod

 def **new** (self, ****config**):

`"""`

 all config options are described in the config module. most relevant are:
 db backend to use, credentials to access db.

 Connect with db

`"""`

 def **update** (self, sql, data):

`"""`

 @sql: str

 @data: list or tuple

 execute sql query with parameters in data. Parameters are marked with “%s” in sql
 `"""`

Sequence Diagrams

This diagram describes what happens when a new tweet is received from the stream and is saved in the data base without any error

