```
import cx_Oracle
import requests
import getpass
import logging
import traceback
from collections import defaultdict, namedtuple
from datetime import datetime

import arrow
import six
from dateutil.tz import tzlocal
from gevent import monkey
from gevent.pool import Pool

import core.services.timeseries.classes.com.pimco.dataservices.timeseries as ts
import etl
from core.db.sql_db import SqlDb
from core.rest.client import JSON_SERIALIZER_DW_ENC
from core.services.timeseries.ts_proxy import TsProxy
from etl.core.da_config import get_env
from etl.core.db import get_db_creds
from etl.core.util import parse_args
from etl.core.util_env import get_cdc_info
from etl.repo.pim_da.dats_series_tss_meta import DatsSeriesTssMetaRepo
from etl.repo.pim_da.dats_series_tss_reg import DatsSeriesTssRegRepo
from etl.repo.pim_da.vtss_dats_series_reg import VtssDatsSeriesRegRepo


def get_tss_base_url():

    # for testing purposes, lets stick to the same server. When committing, comment out the
below line
    # return 'http://devpmwsv7:61000/timeseries/v1'

    return get_env()

class QueuerAgent:

    def _init__(self):
        logging.info('QueuerAgent')
        repo = etl.repo.OraPimRepo()
        cred = get_db_creds(_repo.server, repo.vendor)
```

```
        self.cdc = get_cdc_info()
        self.sql_db = SqlDb(_cred.server, cred.vendor, user=_cred.get('user'),
password=_cred.get('passwd'))
        self.ts_proxy = TsProxy(base_url=_get_tss_base_url())
        self.USERNAME = getpass.getuser()
        monkey.patch_socket()

    def getrequests(self):
        sql = """
            SELECT  FROM pm_own.pl_bbg_batch_vw
            WHERE status_code = :status_code"""
        params = dict(status_code=1)
        data = self.sql_db.query(sql, params)
        return data

    def getrequestobject(self,objdata, result_series):
        obj = dict()
        obj['requestor_code'] = 'BT.DEV'    #objdata[0]
        obj["request_description"] = 'TEST'  #objdata[1]
        obj["program_code"] = objdata[1]
        obj["interface_code"] = objdata[2]
        obj["response_format_code"] = 'HORIZONTAL'    #objdata[4]
        items_list = []
        request_fields_list = []
        for i in result_series:
            element = dict()
            element['yellow_key'] = i[5]
            element['ticker'] = i[4]
            element['tag'] = i[3]
            items_list.append(element)
            request_fields_list.append(i[6])
        obj['request_data_items'] = items_list
        obj['request_fields'] = request_fields_list
        return obj


    def update_request(self,batch_id, bt_request_id, status_code):
        sql = """
            UPDATE pm_own.pl_bbg_batch SET bt_request_id =:a ,status_code= :b WHERE
batch_id = :c """
        params = dict(a= bt_request_id, b= status_code, c= batch_id)
        self.sql_db.query(sql, params)
```

```
    def run()
        result = getrequests()
        for i in result:
            sql = """
            SELECT  FROM pm_own.pl_bbg_batch_series_vw where batch_id = :batch_id"""
            params = dict(batch_id=i[0])
            result_batch = self.sql_db.query(sql, params)
            obj = getrequestobject(i, result_batch)
            print obj
            response = {
                "status": "SUCCESS",
                "body": {
                    "is_error": False,
                    "status_date": "2018-03-01 14:59:32",
                    "request_id": 1021,
                    "progression_url":
"http://ptpcore-webfarm4-dev:50000/workshop/service/da/bbg_transport/check_status/1021",
                    "request_status": "INITIAL",
                    "response_file_info": [],
                    "data_file_path": "null"
                }
            }

    # response = requests.post('http://ptp-dev/workshop/service/da/bbg_transport/request_data/',
data=obj)
    # update_request(i[0], response.request_id, 2)
    # print(i[0])
    # print response['body']['request_id']

            update_request(i[0], response['body']['request_id'], 2)
             # queuerequests(result)

if _name_ == '__main__':
        agent = QueuerAgent()
        agent.run()
```