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()