import getpass  
import logging  
  
import pandas as pd  
from gevent import monkey  
  
from core.db.sql\_db import SqlDb  
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\_env import get\_cdc\_info  
from etl.repo import OraPimRepo  
from etl.repo.fnd\_cfdw.d\_ts\_series import PLBbgRepo  
from etl.repo.fnd\_cfdw.stgp\_f\_ts\_series import PLBbgSeriesRepo  
  
  
  
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 = 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):  
 data = repo.query.filter( status\_code== 1)  
 return data  
  
  
 def getrequestobject(self, objdata, result\_series):  
 obj = dict()  
 obj['requestor\_code'] = 'DA.PIMCOLIVE.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'] = list(set(request\_fields\_list))  
 return obj  
  
 def update\_request(self, batch\_id, bt\_request\_id, bt\_status\_code, obj, status\_code):  
 update\_row=repo.query.filter( status\_code== 1,batch\_id=batch\_id)  
 update\_row.status\_code=2  
 update\_row.bt\_request\_id=bt\_request\_id  
 update\_row.bt\_status\_code=bt\_status\_code  
 update\_row.commit()  
  
 def run(self):  
 result = self.getrequests()  
 for i in result:  
 result\_batch = repo.query.filter( batch\_id== i[0])  
 obj = self.getrequestobject(i, result\_batch)  
 print obj  
 response = {  
 "status": "SUCCESS",  
 "body": {  
 "is\_error": False,  
 "status\_date": "2018-03-01 14:59:32",  
 "request\_id": 200,  
 "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)  
 print(response['body']['request\_status'])  
 self.update\_request(i[0], response['body']['request\_id'], str(obj), str(response['body']['request\_status']), 2)  
  
  
 # queuerequests(result)  
  
  
if \_name\_ == '\_\_main\_\_':  
 agent = QueuerAgent()  
 agent.run()