import getpass  
import logging  
  
from sqlalchemy import or\_  
  
from etl.core.util import uri\_get  
from etl.repo.pim\_pm.pl\_bbg\_batch import PlBbgBatchRepo  
  
BASE\_URL = 'http://ptp-dev/workshop/service/da/bbg\_transport/'  
class FetcherAgent:  
 """  
  
  
 """  
 def \_\_init\_\_(self):  
 logging.info('FetcherAgent')  
 self.USERNAME = getpass.getuser()  
  
   
 def get\_requests(self,repo):  
 """  
  
 :param repo:  
 :return:  
 """  
 model = repo.model  
 data = repo.query.filter(  
 or\_(model.batch\_status\_code == 'SENT\_TO\_BT', model.batch\_status\_code == 'PENDING')).all()  
 return data  
  
 def get\_request\_status\_by\_url(self,obj):  
 """  
  
 :param obj:  
 :return:  
 """  
 response = uri\_get(BASE\_URL + 'check\_request\_status' + str(obj.bt\_request\_id))  
 return response  
  
  
 def update\_request(self,batch\_id, data\_file\_path, bt\_request\_id, status\_code, repo):  
 """  
  
 :param batch\_id:  
 :param data\_file\_path:  
 :param bt\_request\_id:  
 :param status\_code:  
 :param repo:  
 :return:  
 """  
 model = repo.model  
 update\_row = repo.query.filter(model.bt\_request\_id == bt\_request\_id, model.batch\_id == batch\_id).all()  
 update\_row[0].batch\_status\_code = status\_code  
 update\_row[0].bt\_response\_file\_path = data\_file\_path  
 repo.save(update\_row)  
  
 def run(self):  
 """  
  
 :return:  
 """  
 result = self.get\_requests(PlBbgBatchRepo())  
 for i in result:  
 response = self.get\_request\_status\_by\_url(i)  
 self.update\_request(i.batch\_id, response['data\_file\_path'],  
 response['request\_id'], response['request\_status'], PlBbgBatchRepo())  
  
  
if \_name\_ == '\_\_main\_\_':  
 agent = FetcherAgent()  
 agent.run()