"""

"""

import getpass

import logging as log

import os

from shutil import copyfile

from sqlalchemy import or\_

from etl.repo.fnd\_cfdw import VEtlSourceMergedRepo

from etl.repo.pim\_pm.pl\_bbg\_batch import PlBbgBatchRepo

md = VEtlSourceMergedRepo.instance.get\_by\_source\_code('PL\_BT\_POLL\_AGENT')

DESTINATION = md.output\_file\_folder

BT\_COMPLETE\_STATUS = ['BBGERROR', 'BTERROR', 'SUCCESS']

class LoaderAgent(object):

"""

"""

def \_\_init\_\_(self):

log.info('LoaderAgent')

self.USERNAME = getpass.getuser()

@staticmethod

def copy\_file(src, dst,program\_code):

if os.path.isdir(dst):

destination\_file=os.path.basename(src)

if program\_code == 'GETDATA':

ext=".inc"

else:

ext=".ful"

name\_ext=os.path.splitext(destination\_file)

destination\_file = name\_ext[0]+ext+name\_ext[1]

dst = os.path.join(dst, os.path.basename(src))

copyfile(src, dst)

log.info("Copying {0} to {1}".format(src, dst))

else:

log.info("Can't copy {0} to destination: {1}".format(src, dst))

return True

@staticmethod

def get\_requests(repo):

"""

:param repo:

:return:

"""

model = repo.model

data = repo.query.filter(

or\_(model.batch\_status\_code == 'SENT\_TO\_BT',

model.bt\_status\_code in BT\_COMPLETE\_STATUS)).all()

return data

@staticmethod

def update\_request(batch\_id, batch\_status\_code, repo):

"""

:param batch\_id:

:param batch\_status\_code:

:param repo:

:return:

"""

model = repo.model

update\_row = repo.query.filter(model.batch\_id == batch\_id).all()

update\_row[0].batch\_status\_code = batch\_status\_code

repo.save(update\_row)

def run(self):

"""

:return:

"""

result = self.get\_requests(PlBbgBatchRepo())

for i in result:

if i.bt\_status\_code == 'SUCCESS':

if self.copy\_file(i.bt\_response\_file\_path.strip(), DESTINATION,i.bbg\_program\_code):

self.update\_request(i.batch\_id, 'BT\_DONE', PlBbgBatchRepo())

if \_\_name\_\_ == '\_\_main\_\_':

agent = LoaderAgent()

log.info('Start of the program')

agent.run()

log.info('End of the program')