import MySQLdb

→Import 用到的module

import requests

from BeautifulSoup import BeautifulSoup

import time

import xlwt

import random

from datetime import date

start = time.ctime() #程式執行開始時間

today = date.today() #取得本日日期

book = xlwt.Workbook(encoding="utf-8")

→Excel 初始化

sheet = book.add\_sheet("python sheet")

sheet.write(0, 0, "Asin")

sheet.write(0, 1, "Question")

sheet.write(0, 2, "Answer")

sheet.write(0, 3, "AnswerDate")

sheet.write(0, 4, "Votes")

sheet.write(0, 5, "AnswerWriter")

sheet.write(0, 6, "QuesAsker")

sheet.write(0, 7, "QuesAskDate")

sheet.write(0, 8, "Each UrL")

db = MySQLdb.connect(host='localhost', user='james',passwd='james123!',db='crawlerdb') #connect mysql db setting

cursor = db.cursor()

cursor.execute("SELECT Asin, QAUrl FROM product where QAUrl !=''") #db query

row = cursor.fetchall() #get db query result

index = 1 #for excel’s index

for rows in row:

#time.sleep(random.randint(3,8))

asin = rows[0]

#ASIN

print asin

#ASIN

try:

r = requests.get(rows[1])

except requests.exceptions.ConnectionError:

r.status\_code = "Connection Refused"

r\_html= r.text.encode('utf8')

soup = BeautifulSoup(r\_html)

numberofQ = None

protect\_timeout = 0

#keep repeating request until getting the data

while numberofQ is None:

protect\_timeout = protect\_timeout + 1

time.sleep(random.randint(3,8))

if protect\_timeout > 20:

break

try:

r = requests.get(rows[1])

r\_html= r.text.encode('utf8')

soup = BeautifulSoup(r\_html)

numberofQ = soup.find('div',{'class':'a-fixed-left-grid-col askPaginationHeaderMessage a-col-left'},{'style':'width:250px;margin-left:-250px;\_margin-left:-125px;float:left;'}).text.encode('utf8').split('of ')[1].split(' q')[0] #總問題數

except:

pass

#=============================================

if protect\_timeout > 20: #request >20 都失敗->skip

sheet.write(index,0,asin)

index = index + 1

continue

else:

print "test2"

#sub soup circle

questions = soup.findAll('div',{'class':'a-fixed-left-grid-col a-col-right'},{'style':'padding-left:0%;\*width:99.6%;float:left;'})   
 #本頁全部的問題資料

num = int(numberofQ) #總頁數

#convert questions to int, calculate the pages of questions

last = num%10

quotient = num/10 #每頁可存10筆所以用10除

if quotient is 0:

last = 0 #餘數

if last > 0:

quotient = quotient + 1 #頁數(商數)

print quotient

#================================================

#將問題link存入mylist,因為有/help與重複頁面所以用not in 處理

mylist=[]

for run in questions:

nestedQ = run.find('a',{'class':'a-link-normal'})

if nestedQ is not None:

if not nestedQ['href'] in mylist:

if '/help' not in nestedQ['href']:

mylist.append(nestedQ['href'])

r.close()

#l2=[]

#[l2.append(i) for i in mylist if not i in l2]

#http://blog.csdn.net/jiedushi/article/details/6769673

for i in range(0,len(mylist)):

time.sleep(random.randint(3,8))

#===pri===

asker = ''

question= ''

askdate= ''

#===pri===

#===sub===

answerDate=''

answer=''

author=''

vote=''

#===sub===

print "========Each Question======="

print mylist[i]

try:

sub\_r = requests.get("http://www.amazon.com"+mylist[i])

except requests.exceptions.ConnectionError:

sub\_r.status\_code = "Connection Refused"

subr\_html= sub\_r.text.encode('utf8')

sub\_soup = BeautifulSoup(subr\_html)

#get each quesion & answer info.

question = sub\_soup.find('meta',{'name':'title'})

if question is not None:

question = question['content'].split('Answers: ')[1].encode('utf8')

askerData = sub\_soup.find('div',{'class':'cdAuthorInfoBlock'}).text.encode('utf8').split('asked by')[1].split('on ')

asker = askerData[len(askerData)-2]

askdate = askerData[len(askerData)-1]

answers = sub\_soup.findAll('div',{'class':'cdMessageInfo'})

sub\_r.close()

for each in answers:

subAnswer = each.find('span',{'style':'display:block'})

if subAnswer.text.encode('utf8') is '':

xcode = subAnswer['id'].split('\_')[1]

subAnswer = each.find('span',{'id':'long\_'+xcode})

answer = subAnswer.text.encode('utf8')

else:

answer = subAnswer.text.encode('utf8')

Authordata = each.find('div',{'class':'answerAuthor'})

if Authordata is not None:

Authordata = Authordata.text.encode('utf8').split('answered on ')

author = Authordata[0]

answerDate = Authordata[1].split('&')[0]

voteinfo = each.find('span',{'class':'votingInfo'})

if voteinfo is not None:

vote = voteinfo.text.encode('utf8')[0]#.split('.')[0]

if vote is 'D': #表示沒人投票

vote = '0'

print "----------------"

print "Question:"+question

print "Answer:"+answer

print vote

print "Writer:"+author

print "AnswerDate:"+answerDate

print "Asker:"+asker

print "AskDate:"+askdate

print askerData

sheet.write(index,0,asin)

sheet.write(index,1,question)

sheet.write(index,2,answer)

sheet.write(index,3,answerDate)

sheet.write(index,4,vote)

sheet.write(index,5,author)

sheet.write(index,6,asker)

sheet.write(index,7,askdate)

sheet.write(index,8,"http://www.amazon.com"+mylist[i])

book.save("QA"+str(today)+".xls")

cursor.execute("INSERT IGNORE INTO raw\_qa(ID, Asin, AmazonID, Question, Ans, Date, votes, Writer, Asker, AskDate)VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",('',asin,'',question,answer,answerDate,vote,author,asker,askdate))

db.commit()

index = index + 1

print "----------------"

#request the last question pages ...

if quotient is 0: #表示沒有第2頁了

pass

else:

for page in range(2,quotient+1): #從2頁開始依序執行

time.sleep(random.randint(3,8))

print `page`

try:

r = requests.get("http://www.amazon.com/ask/questions/asin/"+asin+"/"+`page`+"/ref=ask\_ql\_qlh\_hza")

except requests.exceptions.ConnectionError:

r.status\_code = "Connection Refused"

r\_html= r.text.encode('utf8')

soup = BeautifulSoup(r\_html)

numberofQ = soup.find('div',{'class':'a-fixed-left-grid-col askPaginationHeaderMessage a-col-left'})

questions = soup.findAll('div',{'class':'a-fixed-left-grid-col a-col-right'},{'style':'padding-left:0%;\*width:99.6%;float:left;'})

reslist=[]

for run in questions: #link存入reslist

nestedQ = run.find('a',{'class':'a-link-normal'})

if nestedQ is not None:

if not nestedQ['href'] in reslist:

if '/help' not in nestedQ['href']:

reslist.append(nestedQ['href'])

r.close()

#close r

for i in range(0,len(reslist)):

time.sleep(random.randint(3,8))

#===pri===

asker = ''

question= ''

askdate= ''

#===pri===

#===sub===

answerDate=''

answer=''

author=''

vote=''

#===sub===

print "========Each Question======="

print reslist[i]

try:

sub\_r = requests.get("http://www.amazon.com"+reslist[i])

except:

pass

subr\_html= sub\_r.text.encode('utf8')

sub\_soup = BeautifulSoup(subr\_html)

#get each quesion & answer info.

question = sub\_soup.find('meta',{'name':'title'})['content'].split('Answers: ')[1]

askerData = sub\_soup.find('div',{'class':'cdAuthorInfoBlock'}).text.encode('utf8').split('asked by')[1].split('on ')

asker = askerData[len(askerData)-2]

askdate = askerData[len(askerData)-1]

answers = sub\_soup.findAll('div',{'class':'cdMessageInfo'})

sub\_r.close()

#close the sub\_r

for each in answers:

subAnswer = each.find('span',{'style':'display:block'})

if subAnswer.text.encode('utf8') is '':

xcode = subAnswer['id'].split('\_')[1]

subAnswer = each.find('span',{'id':'long\_'+xcode})

answer = subAnswer.text.encode('utf8')

else:

answer = subAnswer.text.encode('utf8')

Authordata = each.find('div',{'class':'answerAuthor'})

if Authordata is not None:

Authordata = Authordata.text.encode('utf8').split('answered on ')

author = Authordata[0]

answerDate = Authordata[1].split('&')[0]

voteinfo = each.find('span',{'class':'votingInfo'})

if voteinfo is not None:

vote = voteinfo.text.encode('utf8')[0]#.split('.')[0]

if vote is 'D':

vote = '0'

print "----------------"

print "Question:"+question

print "Answer:"+answer

print vote

print "Writer:"+author

print "AnswerDate:"+answerDate

print "Asker:"+asker

print "AskDate:"+askdate

print askerData

sheet.write(index,0,asin)

sheet.write(index,1,question)

sheet.write(index,2,answer)

sheet.write(index,3,answerDate)

sheet.write(index,4,vote)

sheet.write(index,5,author)

sheet.write(index,6,asker)

sheet.write(index,7,askdate)

sheet.write(index,8,"http://www.amazon.com"+reslist[i])

book.save("QA"+str(today)+".xls") #excel檔名

cursor.execute("INSERT IGNORE INTO raw\_qa(ID, Asin, AmazonID, Question, Ans, Date, votes, Writer, Asker, AskDate)VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",('',asin,'',question,answer,answerDate,vote,author,asker,askdate))

db.commit() #寫入至database

index = index + 1

print "----------------"

print "Start : %s" % start

print "End : %s" % time.ctime() #印出開始時間與結束時間

#=============================================