现代程序设计第十四次作业

所有作业上传至github，地址为'https://github.com/hxx12138/python_mp2021_project.git'

作业要求

以Bilibili热榜为例，练习协程与非关系数据库的使用

库的导入

```
import pymongo
from pymongo import collection
import pandas as pd
import datetime
import pickle
import os
import time
import requests as rs
import asyncio
import aiohttp
import random
from email.header import Header
from email.mime.text import MIMEText
from smtplib import SMTP_SSL
```

作业要求1

获取某一分区视频的排行榜信息，并从中解析出前 10000 条视频信息。可使用协程爬取数据。注意协程应用的方式：发送http请求后，可以跳转到数据处理的函数中，而不要跳到发送另一个http请求的函数中，因为由于b站的风控较为严格，短时间大量的请求可能会导致ip被封禁。

```
# 通过修改学长的代码实现爬虫

class BiliHotCrawler():
    def __init__(self,cate,limit=10000):
        self.cate_id = cate
        self.limit = limit
        self.page = 1
        self.pagesize = 100
        self.bulid_param()
        self.url = BASE_URL

    def bulid_param(self):
```

```
        self.params = {
            'main_ver':'v3',
            'search_type':'video',
            'view_type':'hot_rank',
            'order':'click',
            'cate_id':self.cate_id,
            'page':self.page,
            'pagesize':self.pagesize,
            'time_from':20211117,
            'time_to':20211124
        }

    async def get_resp(self):
        time.sleep(random.choice(sleep_choice))
        self.bulid_param()
        async with aiohttp.ClientSession() as session:
            async with session.get(url=self.url,params=self.params) as response:
                self.resp = await response.read()
        self.page += 1

    def save_resp(self):
        path = f"{SAVE_PATH}/{self.cate_id}.csv"
        with open(path,"a") as f:
            self.resp = eval(self.resp.decode('utf-
8').replace('null','None').replace('false',"False").replace("true","True").replace("\n"
," "))
            for item in self.resp["result"]:
                title = str(item["title"]).replace('\n','。').replace(',',' ')
                description = str(item["description"]).replace('\n','。').replace(',','
')
                # 分类 排名 bv号 时长 播放量 弹幕 标题 封面 评论 收藏 描述 直链
                f.write(f'{self.cate_id},{item["rank_offset"]},{item["bvid"]},
{item["duration"]},{item["play"]},{item["video_review"]},{title},{item["pic"]},
{item["review"]},{item["favorites"]},{description},{item["arcurl"]}'.replace("\n","
").encode('utf-8', 'replace').decode('utf-8'))
                f.write("\n")

    def log(self,isend=False):
        print(f"\rCrawlering {self.cate_id} {(self.page -
1)*self.pagesize}/{self.limit}",end="")
        if isend:
            print()

    async def start(self):
        while self.pagesize * self.page <= self.limit:
            self.log()
            await self.get_resp()
            self.save_resp()
        self.log(isend=True)
```

```python
class BHCFactory():
    def __init__(self,cate_list):
        self.cate_list = cate_list

    def produce(self): # 相当于开了多个协程去分别爬取每个板块的热榜
        loop = asyncio.get_event_loop()
        tasks = [BiliHotCrawler(i).start() for i in self.cate_list]
        loop.run_until_complete(asyncio.wait(tasks))


class BiliNoticeMail:
    def __new__(cls, *args, **kwargs):
        if not hasattr(BiliNoticeMail, '_instance'):
            BiliNoticeMail._instance = object.__new__(cls, *args, **kwargs)
        return BiliNoticeMail._instance

    def __init__(self):
        self.server = SMTP_SSL(SMTP_SERVER, PORT)
        self.message = None
        pass

    def build_ready_message(self):
        self.message = MIMEText('下载完成', 'plain', 'utf-8')
        self.message['From'] = Header(FROM_ADDR, 'utf-8')
        self.message['To'] = Header(TO_ADDR, 'utf-8')
        self.message['Subject'] = Header('BiliReminder:Ready', 'utf-8')

    def notice(self):
        if self.message is None:
            raise ValueError('Message is none!')
        self.server.login(FROM_ADDR, PASSWD)
        self.server.sendmail(FROM_ADDR, [TO_ADDR], self.message.as_string())
        self.message = None
        self.server.quit()

    @classmethod
    def send_ready_mail(cls):
        noticer = cls()
        noticer.build_ready_message()
        noticer.notice()
        print("发信成功，请查收。")
        return noticer

def main():
    BHCFactory(CATE_LIST).produce()
    BiliNoticeMail.send_ready_mail()
```

```
if __name__ == '__main__':
    main()
```

作业要求2

将视频信息存入MongoDB数据库，并记录此视频当时的排名和数据创建时间。

```
client =
pymongo.MongoClient("mongodb+srv://hxx:hexihexiang2000@cluster0.amu8k.mongodb.net/stude
nt?retryWrites=true&w=majority")

db = client['info']
collections_1 = db['week_1']
collections_2 = db['week_2']

db_list = client.list_database_names()
print(db_list)


data_list_1 = [{'分区号':0, '排名':0, 'BV号':0, '时长':0, '播放量':0, '弹幕':0, '标题':0, '封
面':0, '评论':0, '收藏':0, '描述':0, '直链':0, '时间':0} for i in range(10001)]
df_1 = pd.read_csv('week_1/21.csv')
title_1 = list(df_1)
print(title_1)
for i in range(len(title_1)):
    print(f'the no.{i} col has completed.')
    for j in range(len(list(df_1[title_1[i]]))):
        data_list_1[j][title_1[i]] = list(df_1[title_1[i]])[j]
#print(data_list_1)

for i in range(len(data_list_1)):
    print(f'insert no.{i+1}')
    date = datetime.datetime.now()
    data_list_1[i]['时间'] = date
    doc = collections_1.insert_one(data_list_1[i])
x = collections_1.find_one()
print(x)


data_list_2 = [{'分区号':0, '排名':0, 'BV号':0, '时长':0, '播放量':0, '弹幕':0, '标题':0, '封
面':0, '评论':0, '收藏':0, '描述':0, '直链':0, '时间':0} for i in range(10001)]
df_2 = pd.read_csv('week_2/21.csv')
title_2 = list(df_2)
print(title_2)
for i in range(len(title_2)):
    print(f'the no.{i} col has completed.')
    for j in range(len(list(df_2[title_2[i]]))):
```

```
        data_list_2[j][title_1[i]] = list(df_2[title_2[i]])[j]
#print(data_list_2)

for i in range(len(data_list_2)):
    print(f'insert no.{i+1}')
    date = datetime.datetime.now()
    data_list_2[i]['时间'] = date
    doc = collections_2.insert_one(data_list_1[i])
x = collections_2.find_one()
print(x)
```

作业要求3

爬取同一分区一周之后的热榜，并从中解析出前 10000 条视频信息。对比两次结果并更新数据库，要求第一次结果需从数据库中取出。根据以下规则更新数据库：对于仅在第二次排行榜中的视频，将信息存入数据库；对于仅在第一次排行榜中的视频，将信息从数据库中删除；对于两次排行榜都存在的视频，更新数据库，保留创建时间，增加更新时间字段，并更新排名。

```
week_1_list = collections_1.find()
'''with open('week_1/week_1_origin.pkl','wb') as f:
    pickle.dump(list(week_1_list),f)'''

i = 0
week_2_list = collections_2.find()
for info_2 in week_2_list:
    print(f'the no.{i+1} has completed.')
    i += 1
    bv = info_2['BV号']
    query = {'BV号':bv}
    if (collections_1.find(query)):
        update_dict = {'$set':collections_1.find(query)[0]}
        collections_1.update_one(query,update_dict)
    else:
        collections_1.insert_one(info_2)

week_1_list = collections_1.find()
'''with open('week_1/week_1_update.pkl','wb') as f:
    pickle.dump(list(week_1_list),f)'''
```

代码测试

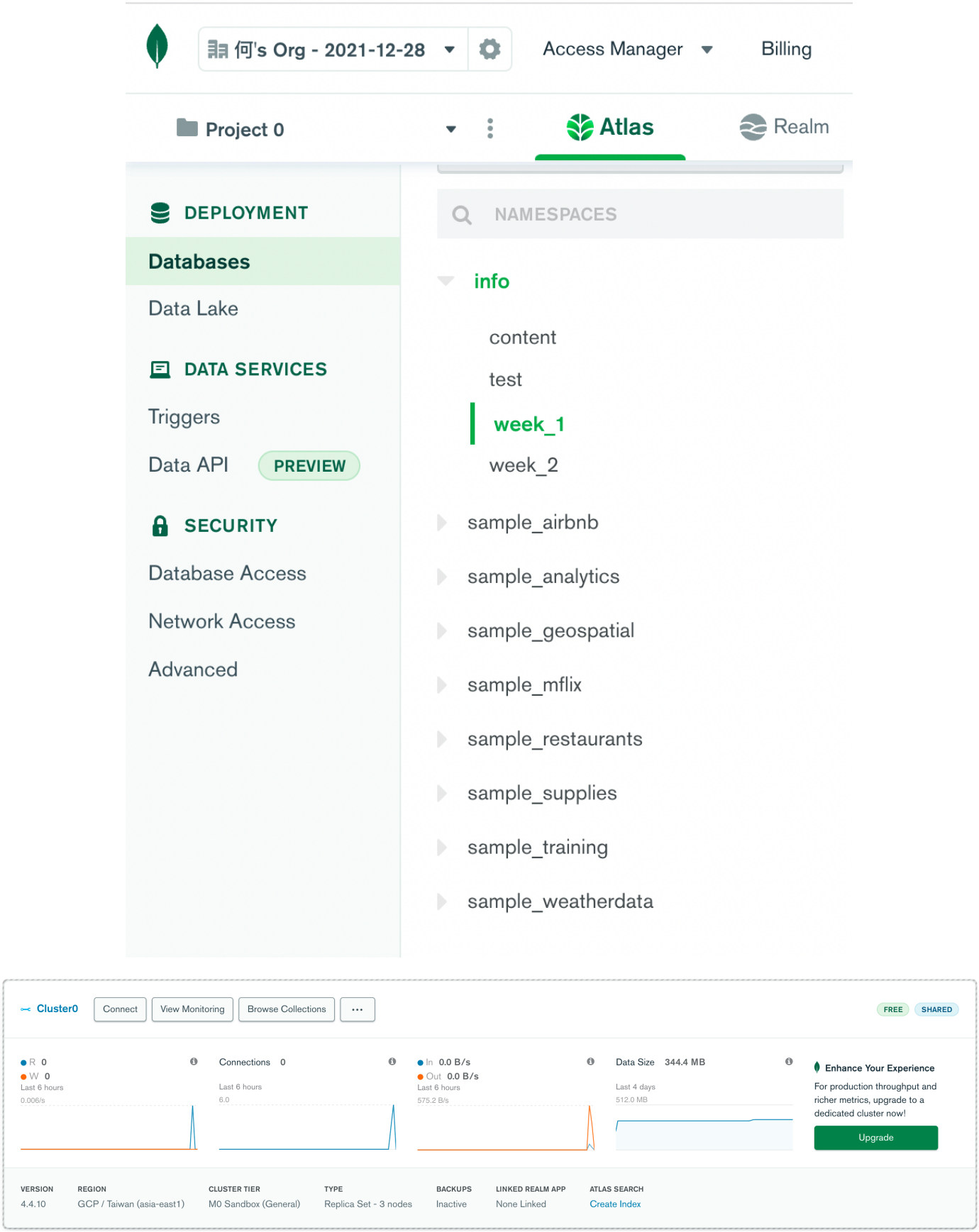
```
import pymongo
from pymongo import collection

client =
pymongo.MongoClient("mongodb+srv://hxx:hexihexiang2000@cluster0.amu8k.mongodb.net/stude
nt?retryWrites=true&w=majority")

db = client['info']
collections_1 = db['week_1']
#collections_2 = db['week_2']

doc_list_1 = collections_1.find().sort('排名')
for doc in doc_list_1:
    if doc['排名'] != 0:
        print(doc)
```

运行结果


何's Org - 2021-12-28
Access Manager
Billing
Project 0
Atlas
Realm
DEPLOYMENT
Databases
Data Lake
DATA SERVICES
Triggers
Data API
PREVIEW
SECURITY
Database Access
Network Access
Advanced
NAMESPACES
info
content
test
week_1
week_2
sample_airbnb
sample_analytics
sample_geospatial
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata

Cluster0
Connect
View Monitoring
Browse Collections
FREE
SHARED
R 0
W 0
Last 6 hours
Connections 0
Last 6 hours
In 0.0 B/s
Out 0.0 B/s
Last 6 hours
Data Size 344.4 MB
Last 4 days
Enhance Your Experience
For production throughput and richer metrics, upgrade to a dedicated cluster now!
Upgrade
VERSION
4.4.10
REGION
GCP / Taiwan (asia-east1)
CLUSTER TIER
M0 Sandbox (General)
TYPE
Replica Set - 3 nodes
BACKUPS
Inactive
LINKED REALM APP
None Linked
ATLAS SEARCH
Create Index

QUERY RESULTS **1-20 OF MANY**


```
_id: ObjectId("61cea89f2e078c18ae584665")
分区号: "21"
排名: "1"
BV号: "BV1QU4y1M7aq"
时长: 232
播放量: 12589769
弹幕: 12047
标题: "外国消防员：早知道中国消防员这样，我就不来了"
封面: "\/\/i0.hdslb.com\/bfs\/archive\/47f3fb15b9c4c79c0106508ed82f4afb72da0e..."
评论: 9195
收藏: 135361
描述: "为了救人 他们愿意飞檐走壁"
直链: "http:\/\/www.bilibili.com\/video\/av676748825"
时间: 2021-12-31T14:52:15.887+00:00
```

```
_id: ObjectId("61cea89f2e078c18ae584666")
分区号: "21"
排名: "2"
BV号: "BV1wh41147oa"
时长: 195
播放量: 9205849
弹幕: 6078
标题: "检讨书的天花板，语文老师说这篇文章太风骚了，我读不出口！"
封面: "\/\/i0.hdslb.com\/bfs\/archive\/2d41c4e9c6fcda97b63a77c89bbf39588e2afb..."
评论: 6779
收藏: 325318
描述: NaN
直链: "http:\/\/www.bilibili.com\/video\/av209232662"
时间: 2021-12-31T14:52:15.979+00:00
```