Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

Пояснительная записка

к семестровой работе

|  |  |  |  |
| --- | --- | --- | --- |
| Автор |  |  | Манукян Арсен Ваганович |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) |
| Группа | ИВТ-365 |  | |
|  | (шифр группы) |  | |
| Проверил |  |  | Коробкин Дмитрий Михайлович |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) |

Волгоград 2020 г.

Содержание

[1 Описание таблиц 3](#__RefHeading___Toc41001462)

[2 Пример заполнения бд записями 4](#__RefHeading___Toc41001462)

[3 Код и его описание 5](#__RefHeading___Toc41001462)

1. Описание таблиц:

header – название новости

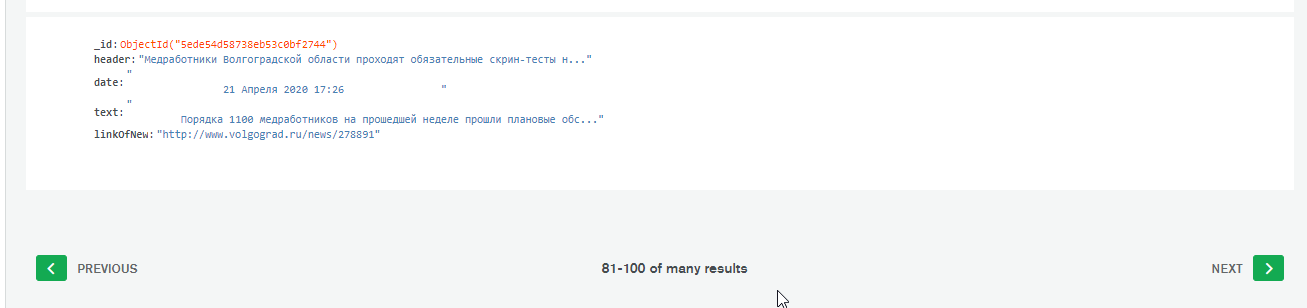
date – дата и время новости

text – основной текст новости

linkOfNew – ссылка на новость

2. Пример заполнения БД записями:





3. Код и его описание:

Подключение к библиотекам

import requests

from pymongo import MongoClient

from bs4 import BeautifulSoup

from multiprocessing import Pool

from datetime import datetime

from tqdm import trange

import sys

from pymongo import MongoClient

Подключение к клиенту mongodb

client = MongoClient("mongodb+srv://root:root@cluster0-xuev7.mongodb.net/sema?retryWrites=true&w=majority")

db=client.sema

news = db.sema

Ловим путь к новостям с помощью самой первой свежей ссылки

mainurl = 'http://www.volgograd.ru/news'

mainpage = requests.get(mainurl).text

mainsoup = BeautifulSoup(mainpage, 'html.parser')

newssoup = mainsoup.find("div", {"class":"col-md-12 news-item"}).find('a').get('href')

Находим число в ссылке, создаем массив, в который поместим все ссылки с 0 по текущую пойманную с помощью цикла

numarticle=""

for char in newssoup:

  if char.isdigit():

    numarticle+=char

all\_links = []

while int(numarticle)>0:

  all\_links.append(mainurl+"/"+str(numarticle))

  numarticle = int(numarticle)-1

Функция main- создаем многопоточность и реализуем ее с созданной функцией make\_all с помощью map

def main():

  start = datetime.now()

  n = len(all\_links)

  listOfList = []

  count\_of\_thread = 10

  for i in range(0, count\_of\_thread):

    listOfList.append(all\_links[int(i\*n/count\_of\_thread):int((i+1)\*n/count\_of\_thread)])

  for i in trange(count\_of\_thread, file=sys.stdout, desc='outer loop'):

    with Pool(count\_of\_thread) as p:

      p.map(make\_all, listOfList[i])

  end = datetime.now()

  total = end - start

  print(total)

if \_\_name\_\_ == '\_\_main\_\_':

  main()

Функция make\_all- получаем переменную soap типа BeautifulSoap и кидаем ее с ссылкой в созданную функцию findcontent

def make\_all(url):

  page = requests.get(url).text

  soup = BeautifulSoup(page, 'html.parser')

  findcontent(url, soup)

Функция findcontent- основная функция с парсингом данных. Заходим в основную ветку, где будем парсить (maincontent), после делаем проверку на наличие новостей (пустых новостей огромное количество и нужно их пропускать), создаем словарь и записываем в него полученные при парсинге данные, после с помощью функции insert\_one вставляем наши словари в mongodb

def findcontent(url, soup):

  maincontent = soup.find("div", {"class":"col-md-9 main-content"})

  if maincontent.find("div", {"class":"news-detail"}) !=None :

    header = maincontent.find('h1').text

    date = maincontent.find("div", {"class":"date"}).text

    texts = maincontent.find("div", {"class":"news-detail"}).find('p').text + maincontent.find("div", {"class":"news-detail"}).find("div", {"id":"full\_text"}).text

    new = {

      "header": header,

      "date": date,

      "text": texts,

      "linkOfNew": url

    }

    news.insert\_one(new)

  else:

    print("я ничего не нашел")