Web与SOA第一次作业

文档

1752854

赵羿昕

# 代码环境：

Python3.7 flask

# 使用库：

from flask import Flask, render\_template, request, session

import requests

import json, urllib

from urllib import parse

from urllib.request import urlopen

from threading import Thread

from azure.cognitiveservices.search.websearch import WebSearchAPI

from azure.cognitiveservices.search.websearch.models import SafeSearch

from msrest.authentication import CognitiveServicesCredentials

# 思路：

使用flask写后端，调用多个api，搜索前端提交的表单中的城市名，把数据返回给千吨啊显示再界面上。

# 调用api描述：

1. 天气信息查询：

发送信息：

params = {

        "key" : appkey, #你申请的key

        "location" : city,

        "unit" : "c"

}

接收信息：

global weather

    if res:

        rst = res["results"][0]["now"]

        weather = rst

1. 生活指数查询：

发送信息：

params = {

        "key" : appkey, #你申请的key

        "location" : city,

}

接收信息：

global live

    if res:

        rst = res["results"][0]["suggestion"]

        live = rst

1. 空气质量查询：

发送信息：

params = {

        "key" : appkey, #你申请的key

        "location" : city,

}

接收信息：

global air

    if res:

        rst = res["results"][0]["air"]["city"]

        air = rst

1. 新闻查询：

发送信息：

params = {

        "showapi\_appid" : "107028",

        "showapi\_sign" : appkey, #你申请的key

        "areaName" : city,

    }

接收信息：

global news

    print(res)

    if res:

        rst = res["showapi\_res\_body"]["pagebean"]["contentlist"]

        n = 0

        prm = []

        for i in rst:

            if n < 6:

                prm.append([i["title"],i["link"].strip('"')])

            else:

                break

            n += 1

        news = prm

1. 必应搜索查询：

发送信息：

try:

         city = city

         web\_data = client.web.search(query=city, offset=0, count=5)

接收信息：

if web\_data.web\_pages.value:

             #print("Webpage Results#{}".format(len(web\_data.web\_pages.value)))

             for i in web\_data.web\_pages.value:

                 if "百科" in format(i.name):

                     webPage = format(i.name)

                     webUrl = format(i.url)

                     webSnippet = format(i.snippet)

             baike = {

                 "page" : webPage,

                 "url" : webUrl,

                 "snippet" : webSnippet

             }

             #print("First web page name: {} ".format(first\_web\_page.name))

             #print("First web page URL: {} ".format(first\_web\_page.url))

         else:

             print("Didn't find any web pages...")

1. 地图：

<script type="text/javascript">

    var value = document.getElementById("cc").innerHTML

        // 百度地图API功能

    var map = new BMap.Map("allmap"); // 创建Map实例

    map.centerAndZoom(value, 12); // 初始化地图,设置中心点坐标和地图级别

    //添加地图类型控件

    map.addControl(new BMap.MapTypeControl({

        mapTypes: [

            BMAP\_NORMAL\_MAP,

            BMAP\_HYBRID\_MAP

        ]

    }));

    map.setCurrentCity(value); // 设置地图显示的城市 此项是必须设置的

    map.enableScrollWheelZoom(true); //开启鼠标滚轮缩放

</script>

# 核心函数：

使用多线程调用api返回数据并重组成data发送给前端，由前端接受并显示

@app.route('/', methods=["GET","POST"])

def hello\_world():

    Data = {"city" : "上海"}

    if request.method == "POST":

        form\_data = request.form

        city = form\_data.get("search")

        t1 = Thread(target=request1, args=(appkey1, city))

        t2 = Thread(target=request2, args=(appkey1, city))

        t3 = Thread(target=request3, args=(appkey1, city))

        t4 = Thread(target=request4, args=(" ", city))

        t5 = Thread(target=web\_results\_with\_count\_and\_offset, args=(subscription\_key, city))

        t1.start()

        t2.start()

        t3.start()

        t4.start()

        t5.start()

        t1.join()

        t2.join()

        t3.join()

        t4.join()

        t5.join()

        Data = {

            "city" : city,

            "temperature" : weather["temperature"],

            "weather" : weather["text"],

            "air\_aqi" : air["aqi"],

            "air\_quality" : live["air\_pollution"]["brief"],

            "air\_pm25" : air["pm25"],

            "air\_primary" : air["primary\_pollutant"],

            "air\_suggestion" : live["air\_pollution"]["details"],

            "live\_comfort" : live["comfort"]["brief"],

            "news" : news,

            "webPage" : baike["page"],

            "webUrl" : baike["url"],

            "webSnippet" : baike["snippet"],

        }

# 使用方法：

配置好库之后直接run，使用浏览器浏览127.0.0.1：5000进行测试