

# 第九周周报

## 第十九小组 蔡逸文 宋健 李宇星

### 本周工作

完成真实景点数据的录入代码

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
{
  path: '116.33258478972,39.582748999315;116.33270218774,39.582758742465;116.33285158646,39.582762918181;116.33304204553,39.582769161554;116.33385859604,39.582790755665;116.33390432852,39.582792147943'
},
{
  path: '116.33390432852,39.582792147943;116.33390791374,39.582814417585'
},
{
  path: '116.33390791374,39.582814417585;116.33394040939,39.583872869819;116.33399584937,39.582979254894;116.33418181723,39.583890880766;116.33415843918,39.5831158262;116.33434359101,39.583134067254;116.33444338983,39.583282869816;116.33454668896,39.5833168835
87;116.33453835412,39.583368182577;116.33471177853,39.583416401946;116.33475869866,39.583444327841;116.33485293114,39.58354438384;116.33490143963,39.58368865325;116.33498588935,39.583696862229;116.3349841971,39.583889551697;116.33508519226,39.583833968265;116.335118744
82,39.583867364821;116.33514757534,39.583959923195'
},
{
  path: '116.33514757534,39.583959923195;116.33528232117,39.584086358298;116.335417067,39.584040658697;116.33558599924,39.584075447805;116.33561649882,39.584151382896;116.33589185818,39.584282835826;116.33579615192,39.584236832853;116.33587258789,39.58423683285
3;116.3368468978,39.58422858952;116.33688548629,39.584229872809;116.33612211818,39.58422113425;116.33614649187,39.5841735725;116.3361671521,39.584888666958;116.33619499957,39.58397933956'
},
{
  path: '116.33619499957,39.58397933956;116.33709061816,39.58393758393;116.33742567812,39.583956443559;116.3375883222,39.583943916869;116.33771379691,39.583915383843;116.33786764443,39.583895828274;116.33798442414,39.58391391988;116.33807066147,39.58393556574'
},
{
  path: '116.33807066147,39.58393556574;116.3380922288,39.583948437232;116.33811467844,39.583963482831'
},
{
  path: '116.33811467844,39.583963482831;116.3381568988,39.584004462518;116.33824313613,39.584039954771;116.33840932265,39.584069810805;116.3384991532,39.584078926635;116.33852618237,39.584081814413;116.33863380072,39.5840844494043;116.33869228888,39.584088214889'
},
{
  path: '116.33869228888,39.584088214889;116.33869139058,39.584238638326;116.33868959397,39.584312861435;116.33868779736,39.584393414855'
},
  origin: '北京野生动物园-自助购票处',
  dest: '北京野生动物园-随行动物馆'
},
{
  distance: 386,
  duration: 336,
  steps: [
    {
      path: '116.33476849042,39.49889939863;116.33476759211,39.499086877367;116.33475681245,39.498987344368;116.33472986328,39.498917746598;116.3346840497,39.498886548225;116.33461577848,39.498831445266;116.33452864285,39.498815368151;116.33442713432,39.498815368151
116.33431574444,39.498825887836'
    },
    {
      path: '116.33431574444,39.498825887836;116.3342923885,39.498865548225;116.33426983255,39.498981843123;116.33425816814,39.498936538803;116.33423138372,39.498956606096;116.334211541,39.499054854138'
    },
    {
      path: '116.334211541,39.499054854138;116.33425465967,39.499085928166;116.33428520285,39.499072879931;116.33438137155,39.499083759222;116.33431484613,39.499102775913;116.3343281225,39.499148818924;116.33434718513,39.499188756282;116.33434628683,39.49922119236
116.33433358718,39.499262254872;116.3343488388,39.49929887187;116.33432633764,39.49933463819;116.33423848542,39.499386456388;116.33414857487,39.499399988189;116.33381218935,39.499464887283;116.33434848587,39.499528116646;116.3343474481,39.499568627317;116.3348837113
39.499587988606;116.33386889654,39.499619219386;116.3333886684,39.499647823787;116.33337521857,39.49966243989;116.33336713382,39.499682622231;116.333635486,39.49971463686;116.3337521857,39.4997584136'
    },
    {
      path: '116.3337521857,39.4997584136;116.3333958796,39.499841383285'
    },
    {
      path: '116.3333958796,39.499841383285;116.3332988626,39.499851742815'
    },
    {
      path: '116.3332988626,39.499851742815;116.33321891341,39.499947786418;116.33314884558,39.500038958121;116.33388586589,39.500119690214;116.33384815862,39.500189286771'
    },
    {
      path: '116.33384815862,39.500189286771;116.33318932014,39.500265842982;116.33317938797,39.500588418112;116.33319816764,39.500892208055;116.33319645577,39.501052904598;116.33317759136,39.501158863544'
    },
    origin: '北京野生动物园-自助购票处',
    dest: '北京野生动物园-北京国际赛场'
  ]
},
}
```

主要代码如下

```

# encoding:utf-8
from datetime import datetime
import asyncio
import requests
import time
import Scop
ak = "obh0u6Si9EEYanw8WI8x9CNtRqe0FYG4"

async def get_routes(origin_lat: float, origin_lng: float, dest_lat: float, dest_lng: float):
    # 接口地址
    url = "https://api.map.baidu.com/directionlite/v1/walking"

    params = {
        "origin": f"{origin_lat},{origin_lng}",
        "destination": f"{dest_lat},{dest_lng}",
        "ak": ak,
    }

    response = requests.get(url=url, params=params)
    if response:
        data = response.json()["result"]
        res = dict()
        res["distance"] = data["routes"][0]["distance"]
        res["duration"] = data["routes"][0]["duration"]
        steps = []
        for step in data["routes"][0]["steps"]:
            temp = dict()
            temp["path"] = step["path"]
            steps.append(temp)
        res["step"] = steps
        return res

async def create_scop(data):
    # 接口地址
    url = "https://api.map.baidu.com/place/v2/suggestion"

    params = {
        "query": data['name'],
        "region": "北京",
        "city_limit": "true",
        "output": "json",
    }

```

```

        "ak":      ak,

    }

    scop = Scop.Scop(data['name'], data['location']
                     ['lat'], data['location']['lng'])
    response = requests.get(url=url, params=params)
    if response:
        print(response.json())
        for build in response.json()['result']:
            scop.add_building(build['name'], build['location']
                             ['lat'], build['location']['lng'])
    for fro in scop.buildings:
        for to in scop.buildings:
            if fro.lat != to.lat and fro.lng != to.lng: # 坐标不同
                routes = await get_routes(fro.lat, fro.lng, to.lat, to.lng)
                scop.add_route(fro.name, to.name, routes)
    await scop.save()

async def create_city(city_code: str):
    if not str.isdigit(city_code):
        return
    # 接口地址
    url = "https://api.map.baidu.com/place/v2/search"
    # 此处填写你在控制台-应用管理-创建应用后获取的AK
    params = {
        "query":    "景区",
        "tag":      "景区",
        "region":   city_code,
        "output":   "json",
        "ak":       ak,
        "scop":     2,
        "city_limit": True,
    }

    response = requests.get(url=url, params=params)
    if response:
        for scop in response.json()['results']:
            await create_scop(scop)
            time.sleep(0.1)
if __name__ == "__main__":
    async def main():
        await create_city("131")

```

```
# with open("./BaiduMap_cityCode_1102.txt") as file:
#     for city in file.readlines():
#         await create_city(city.split(',')[0])
asyncio.run(main())
print("done")
```

完成导航界面的初步显示（但还存在轨迹坐标不对的问题）

## 下周安排

完善界面显示，修改上述bugsV