

Golang开发实战

运行环境：

Ubuntu 16.04

go1.9.2

任务介绍：

民兵执行任务中，位置会不断更新。设计一个地理信息交互模块，该模块可以：

- 根据任务id获取某次活动中所有人的位置；
- 根据任务id和民兵id获取他的位置；
- 更新某个民兵某次任务中的位置。

解决方案：

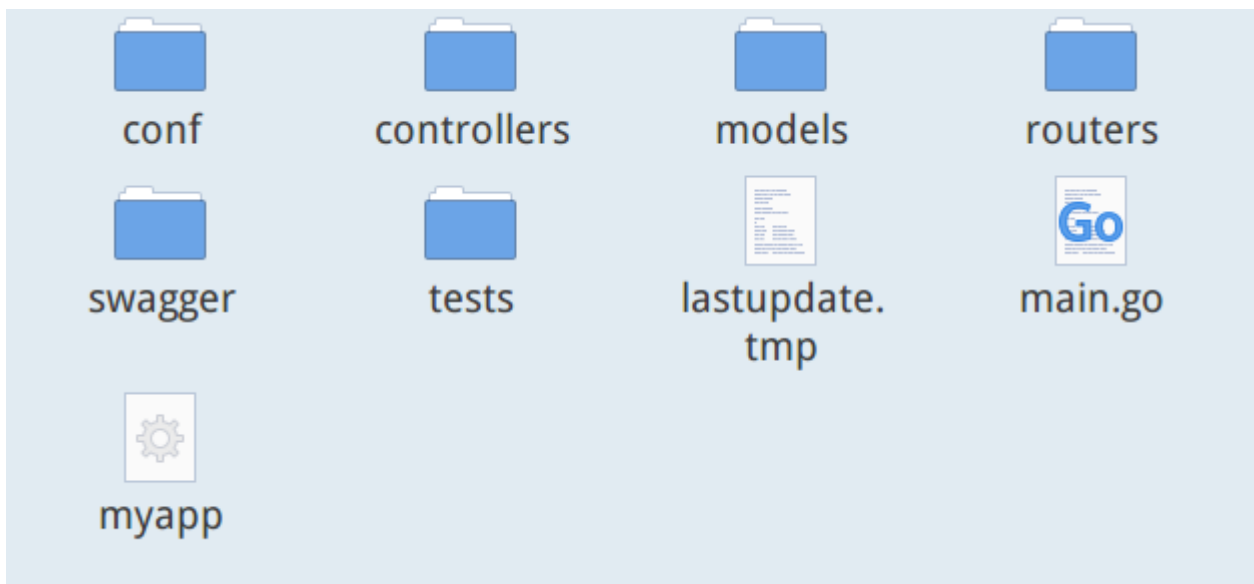
1、在搭建好go环境的前提下，执行以下命令新建一个项目：

```
bee api myapp
```

cd 到myapp下，执行：

```
bee run -gendoc=true -downdoc=true
```

文件目录如下：



在该目录下执行go run main.go即可运行项目。

2、设计三个接口：

①获取某次活动所有人的位置信息（/mbgc/loc/getAllLocs）介绍：根据活动的ID获得该次活动所有人的位置信息。每一条位置信息是一个结构体model，包含 task_id(任务id)，soldier_id(士兵id) 以及经度和纬度。所有地理信息都保存在内存中，每次根据id进行查询。②获取某次活动中某人的位置信息。（/mbgc/loc/getOneLocs）介绍：根据task_id以及soldier_id返回对应的经度和纬度。③实时更新某次活动中某人的位置。

（/mbgc/loc/upOneLocs）介绍：根据task_id以及soldier_id查看该位置信息是否已经存在，若是，更新；否则新建。

3、实现：

model：每条地理位置信息都是一个struct。全部地理位置信息保存在一个结构体slice中，定义插入、查询、更新操作。

```
type Location struct {
    Id int
    Task_id int
    Soldier_id int
    Longitude float64
    Latitude float64
}
```

controller：定义处理客户端请求的各个函数，包括 GetAll（获取所有位置信息），GetById（获取某人位置信息），Post（更新某人位置信息）。各个函数通过调用model中的操作实现。

```
func (u *LocationController) GetAll() {
    ss := models.GetAllLocations()
    u.Data["json"] = ss
    u.ServeJSON()
}
```

router：使用beego的namespace方式绑定路由。

```
func init() {
    ns := beego.NewNamespace("/loc",
        beego.NSNamespace("/location",
            beego.NSInclude(
                &controllers.LocationController{},
            ),
        ),
    )
    beego.AddNamespace(ns)
}
```

4、测试：

利用postman测试如下：

首先上传一个地址，post方法，需要四个参数，task_id代表是哪次任务，soldier_id代表哪个士兵，longitude和latitude代表士兵所在经度和纬度。上传成功后返回成功信息。

POST http://localhost:8090/mbgc/loc/upOneLocs Params Send

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> task_id	5	
<input checked="" type="checkbox"/> soldier_id	1	
<input checked="" type="checkbox"/> longitude	1.888	
<input checked="" type="checkbox"/> latitude	2.33	
New key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 20 ms

Pretty Raw Preview JSON

```
1 {
2   "cnmsg": "成功",
3   "code": 200,
4   "enmsg": "ok"
5 }
```

其次是获取某次任务所有民兵地址，只有一个参数task_id。返回成功信息和所有任务id为task_id的民兵地址。

POST http://localhost:8090/mbgc/loc/getAllLocs Params Send

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> task_id	5	
<input type="checkbox"/> soldier_id	1	
<input type="checkbox"/> longitude	1.888	
<input type="checkbox"/> latitude	2.33	
New key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 16 ms

Pretty Raw Preview JSON

```
1 {
2   "cnmsg": "成功",
3   "code": 200,
4   "data": [
5     {
6       "task_id": 5,
7       "soldier": 1,
8       "longitude": 1.888,
9       "latitude": 2.33
10    }
11  ],
12   "enmsg": "ok"
13 }
```

最后是获取某次活动某人的地址，参数是task_id和soldier_id。返回成功信息和位置信息。

POST

http://localhost:8090/mbgc/loc/getOneLocs

Params

Send

form-data

x-www-form-urlencoded

raw

binary

	Key	Value	Description
<input checked="" type="checkbox"/>	task_id	5	
<input checked="" type="checkbox"/>	soldier_id	1	
<input type="checkbox"/>	longitude	1.888	
<input type="checkbox"/>	latitude	2.33	
	New key	Value	Description

Body

Cookies

Headers (4)

Test Results

Status: 200 OKTime: 15 ms

Pretty

Raw

Preview

JSON

1

{

2

"cnmsg": "成功",

3

"code": 200,

4

"data": {

5

"task_id": 5,

6

"soldier": 1,

7

"longitude": 1.888,

8

"latitude": 2.33

9

},

10

"enmsg": "ok"

11

}