

服务计算——goAgenda

1. 安装使用 cobra
2. goAgenda知识点汇总
 - 2.1 JSON读写学习
 - 2.1.1 编码
 - 2.1.2 解码
 - 2.1.3 JSON转结构体
 - 2.1.4 转换接口
 - 2.2 文件读写
 - 2.3 实例
 - 2.4 StringArray使用
 - 2.5 异常
3. 分析设计
 - 3.1 子命令
 - 3.2 agenda help
 - 3.3 agenda user register
 - 3.4 agenda user login
 - 3.5 agenda user logout
 - 3.6 agenda user lookup
 - 3.7 agenda user delete
 - 3.8 agenda meeting create
 - 3.9 agenda meeting addUser
 - 3.10 agenda meeting deleteUser
 - 3.11 agenda meeting lookup
 - 3.11 agenda meeting cancel
 - 3.12 agenda meeting exit
 - 3.13 agenda meeting clear
4. 数据结构定义
 - 4.1 用户(user)
 - 4.2 会议(conference)
5. 程序测试
 - 5.1 agenda根命令
 - 5.2 help子命令
 - 5.3 user子命令
 - 一、lookup
 - 二、login
 - 三、register
 - 四、delete
 - 5.4 meeting子命令
 - 一、lookup
 - 二、addUser
 - 三、deleteUser
 - 六、cancel
 - 七、create
 - 八、exit
 - 九、clear
6. 项目组内成员与分工

- 7. 源代码
- 8. 参考资料

服务计算——goAgenda

1. 安装使用 cobra

根据老师的指令：

- `go get -v github.com/spf13/cobra/cobra`

随后会遇到以下的情况(提示报错信息)：

Fetching <https://golang.org/x/sys/unix?go-get=1>

https fetch failed: Get <https://golang.org/x/sys/unix?go-get=1>: dial tcp 216.239.37.1:443: i/o timeout

到这里我们有两个解决的办法：

1. 进入 `$GOPATH/src/golang.org/x` 文件夹下，使用：
 - git clone <https://github.com/golang/sys.git>
 - git clone <https://github.com/golang/text.git>

这两条指令将远程库的文件克隆到本地的文件夹下

2. [进入该链接](#)，下载所需的文件，解压后复制到本地

随后使用

- `go install github.com/spf13/cobra/cobra`

即可完成 `cobra` 的安装

2. goAgenda知识点汇总

2.1 JSON读写学习

2.1.1 编码

结构体转JSON代码：

```
package main

import (
    "encoding/json"
    "fmt"
)
```

```

type DebugInfo struct {
    Level  string `json:"level,omitempty"` // Level解析为level,忽略空值
    Msg    string `json:"message"`             // Msg解析为message
    Author string `json:"- "`                 // 忽略Author, 或者设为未导出字段
}

func main() {

    dbgInfs := []DebugInfo{
        DebugInfo{"debug", `File: "test.txt" Not Found`, "Cynhard"},
        DebugInfo{"", "Logic error", "Gopher"},
    }

    if data, err := json.Marshal(dbgInfs); err == nil {
        fmt.Printf("%s\n", data)
    }
}

```

自定义结构对MarshalJSON()接口的实现

```

package main

import (
    "encoding/json"
    "fmt"
)

type Point struct{ X, Y int }

func (pt Point)MarshalJSON() ([]byte, error) {
    return []byte(fmt.Sprintf(`{"X":%d,"Y":%d}`, pt.X, pt.Y)), nil
}

func main() {
    if data, err := json.Marshal(Point{50, 50}); err == nil {
        fmt.Printf("%s\n", data)
    }
}

```

2.1.2 解码

解码主要使用的是JSON包的函数 `func Unmarshal(data []byte,v interface{}) error`

```

package main

```

```
import (
    "encoding/json"
    "fmt"
)

func main() {
    data := `[{"Level":"debug","Msg":"File: \"test.txt\" Not Found"},` +
        `[{"Level":"","Msg":"Logic error"}]`

    var dbgInfos []map[string]string
    json.Unmarshal([]byte(data), &dbgInfos)

    fmt.Println(dbgInfos)
}
```

2.1.3 JSON转结构体

与编码一样，JSON是通过反射机制来实现解码的，所以结构必须导出所转换的字段，不导出的字段不会被JSON包解析。另外解析的时候不区分大小写。

```
package main

import (
    "encoding/json"
    "fmt"
)

type DebugInfo struct {
    Level string
    Msg string
    author string // 未导出字段不会被json解析
}

func (dbgInfo DebugInfo) String() string {
    return fmt.Sprintf("{Level: %s, Msg: %s}", dbgInfo.Level, dbgInfo.Msg)
}

func main() {
    data := `[{"level":"debug","msg":"File Not Found","author":"Cynhard"},` +
        `[{"level":"","msg":"Logic error","author":"Gopher"}]`

    var dbgInfos []DebugInfo
    json.Unmarshal([]byte(data), &dbgInfos)

    fmt.Println(dbgInfos)
}
```

结构体也可以与编码的时候一样设置结构体字段标签

2.1.4 转换接口

和编码的时候类似，解码使用接口Unmarshaler，需要实现 `UnmarshalJSON([]byte) error`

下面这个接口没有实现解码算法，只是将参数打印出来

```
package main

import (
    "encoding/json"
    "fmt"
)

type Point struct{ X, Y int }

func (Point) UnmarshalJSON(data []byte) error {
    fmt.Println(string(data))
    return nil
}

func main() {
    data := `{"X":80,"Y":80}`
    var pt Point
    json.Unmarshal([]byte(data), &pt)
}
```

2.2 文件读写

使用ioutil的ReadFile和WriteFile，并使用string()等函数进行类型转换，例程：

```
package main

import(
    "fmt"
    "io/ioutil"
)

func main() {
    //read
    b, err := ioutil.ReadFile("test.log")
    if err != nil {
        fmt.Print(err)
    }
    fmt.Println(b)
    str := string(b)
    fmt.Println(str)
    //write
}
```

```
d1 := []byte("hello\ngo\n")
err := ioutil.WriteFile("test.txt", d1, 0644)
check(err)
}
```

2.3 实例

使用user进行的测试（已经通过）

```
var myuser []entity.User
fmt.Println("JSON decode to structure test1")
jsonStr := `[{"username":"wtysos11","password":"123456","email":"wtysos11"}
{"username":"wtysos11","password":"123456","email":"wtysos11"}]`
json.Unmarshal([]byte(jsonStr), &myuser)
fmt.Println(myuser)

if data, err := json.Marshal(myuser); err == nil {
    fmt.Printf("%s\n", data)
}
```

2.4 StringArray使用

使用 `cmd.Flags().StringArray()` 可以声明一个flag 使用 `cmd.Flags().GetStringArray()` 可以取得这个数组 对于数组而言只有重复使用标签才能够作为不同元素，比如 `-a="ss" -a="tt"`，而同一个标签内都算作是一个字符窜

2.5 异常

```
if 遇到错误
    return errors.New("遇到xx错误")
```

3. 分析设计

3.1 子命令

- 大的方向上要实现 `help`、`user`、`meeting` 三个子命令
 - 用户信息存储在 `curUser.txt` 中，`User` 和 `Meeting` 实体要用 `json` 进行存储
 - 实现日志服务
-

3.2 agenda help

寻求帮助使用子命令 `agenda help`，列出命令说明。此外输入 `agenda help register` 可以列出 `register` 命令的描述

3.3 agenda user register

用户注册使用命令 `agenda user register`

该指令可接受四个参数（必选）：

- `--username/-u`，用户名
- `--password/-p`，密码
- `--email/-e`，邮箱
- `--telephone/-t`，电话

注：

- 此外用户名是唯一的，要查重。
- 邮箱、电话要检查有效性。

反馈：

- 如果登记成功，返回成功注册的信息
 - 如果登记失败，反馈错误信息
-

3.4 agenda user login

用户登录使用子命令 `agenda user login`

- `--username/-u`，用户名
- `--password/-p`，密码

注：

- **用户名与密码**这两个参数为必须

反馈：

- 用户名与密码都正确的时候返回一个成功登录的信息
 - 登录失败，返回登录失败的信息
-

3.5 agenda user logout

用户登出使用子命令 `agenda user logout`

注：

用户登出后只能使用用户注册和用户登录功能且这里命令**不接受参数**

反馈：

- 反馈登出信息
-

3.6 agenda user lookup

在创建的用户中查找某用户使用子命令 `agenda user lookup`

注：

只有在已经登录的状态才可以使用

反馈：

- 如果找到用户，返回已注册的所有用户的用户名、邮箱以及电话信息（打表）
-

3.7 agenda user delete

删除用户使用子命令 `agenda user delete`

注：

- 操作为删除自己的账号，自动注销
- 用户账号删除之后：
 - 以该用户为发起者的会议将会被删除
 - 以该用户为参与者的会议将会从参与者列表中移除该用户。如果此操作造成参与者数量为0，则会议也会被移除。

反馈：

- 返回成功注销的信息
 - 失败要返回注销失败（有什么情况会导致注销失败吗？）
-

3.8 agenda meeting create

创建会议使用子命令 `agenda meeting create`

参数：

- `--start/-s`，开始时间，格式为(YYYY-MM-DD/HH:mm:ss)
- `--end/-e`，结束时间
- `--title/-t`，会议主题
- `--participant/-p`，会议参与者

注：

- 用户无法分身参与多个会议，如果用户已有的会议安排与将要创建的会议在时间上有重叠，则无法创建这个会议。
- 会议主题是唯一的。

反馈：

- 成功返回创建成功提示信息
- 失败返回创建失败相关信息

3.9 agenda meeting addUser

添加会议参与者使用子命令 `meeting addUser -p [Participator] -t [Title]`

参数：

- --title/-t, 会议主题
- --participants/-p, 要添加的用户名（可以为多个）

反馈：

- 成功返回相关信息
- 失败也返回相关信息

3.10 agenda meeting deleteUser

删除会议参与者使用子命令 `meeting deleteUser -p [Participator] -t [Title]`

参数：

- --title/-t, 会议主题
- --participants/-p, 要添加的用户名（可以为多个）

反馈：

- 成功返回相关信息
- 失败也返回相关信息

3.11 agenda meeting lookup

查找会议(仅登录可用)使用子命令 `meeting lookup -s [StartTime] -e [EndTime]`

参数：

- --start/-s, 开始时间
- --end/-e, 结束时间

注：

- 已登录的用户可以查询自己的议程在某一时间段内的所有会议安排

3.11 agenda meeting cancel

取消会议(已经登录的用户可以取消自己发起的某一会议安排)使用子命令 `meeting cancel -t [title]`

参数:

- `--title/-t`, 会议主题

3.12 agenda meeting exit

退出会议(已经登录的用户可以退出自己参与的某一会议安排)使用子命令 `meeting exit -t [title]`

参数:

- `--title/-t`, 会议主题

注:

- 如果此操作造成会议参与者人数为0, 则会议将会被删除

3.13 agenda meeting clear

清空会议(已经登录的用户可以清空自己发起的所有会议安排)使用子命令 `clear`

4. 数据结构定义

4.1 用户(user)

位置: `entity/useroper.go` 中

作用: 定义用户所需要的结构, 以及相关的操作函数

```
type User struct{

    Username string //用户名

    Password string //密码

    Email string // 电子邮箱

    Telephone string // 电话

}
```

4.2 会议(conference)

位置: `entity/conference.go` 中

作用：定义会议所需要的结构，以及相关的操作函数

```
type Meeting struct{

    StartTime string //开始时间

    EndTime string //结束时间

    Title string //会议标题

    UserList []User //参与人数列表
}
```

5. 程序测试

为了可以在任意位置使用agenda命令，而不是进入工作目录后 `go run main.go`，采用先 `go build main.go` 后重命名可执行程序为agenda，然后移动到 `gowork/bin` 中，在每次运行命令时，golang会根据GOPATH的位置，在bin中找到相应的可执行程序，从而在任意位置使用agenda命令。

本次测试的shell为Z shell（zsh）。

根据cmd-design中的需求，对三个子命令分别进行测试，并将相应输出记录到日志中。

5.1 agenda根命令

```
➔ ~ agenda
agenda

This application is build on the cobra library which allows the client to manage the schedule of different
meetings as well as the infomation of participants and sponsors.
To get a detailed usage of command and flags,please input
'agenda help user' or 'agenda help meeting'

Usage:
  agenda [command]

Available Commands:
  help      helo user to do something
  help      Help about any command
  meeting   operation on meeting
  user      help a user to regist

Flags:
  --config string  config file (default is $HOME/.agenda.yaml)
  -h, --help       help for agenda
  -t, --toggle     Help message for toggle

Use "agenda [command] --help" for more information about a command.
```

5.2 help子命令

可以查看user子命令的帮助信息，用法如下：

1. `agenda help user`
2. `agenda user -help`

```
→ ~ agenda help user
agenda help user

1. To add a register:

instruction:    user register -u [UserName] -p [Pass] -e [Email] -t [Phone]

               [Username] register's name
               [Pass] register's Password
               [Email] register's email
               [Phone] register's phone-number

2. To remove a register

instruction:    user delete

(attention: you can delete your account in the database of Agenda)

3. To query user

instruction:    user lookup

(attention: you can query all user's name who has registred)

4. To login

instruction:    user login -u [UserName] -p [PassWord]

               [Username] register's name
               [Pass] register's Password

5. To logout

instruction:    user logout
```

还可以查看用户命令下的其他子命令用法，如 `agenda help user register` 等。

```

→ ~ agenda help user register
agenda help user register

Command : user register
Function: Register a user.
instruction: user register --username/-u [UserName] --password/-p [Pass] --email/-e [Email] --telephone/-t [Phone]
Args :
    [Username]    register's name
    [Pass]        register's Password
    [Email]       register's email
    [Phone]       register's phone-number
→ ~ agenda help user delete
agenda help user delete

Command : user delete
Function: Delete the current user.
instruction: user delete

→ ~ agenda help user lookup
agenda help user lookup

Command : user lookup
Function: Lookup all the information of the users. (Need to login first)
instruction: user lookup

→ ~ agenda help user login
agenda help user login

Command : user login
Function: Login
(PS: If you don't login, you can only use user login and user register commands.)
instruction: user login --username/-s [UserName] --password/-p [Pass]
Args :
    [Username]    register's name
    [Pass]        register's Password

```

```

→ ~ agenda help user logout
agenda help user logout

Command : user logout
Function: Logout
instruction: Logout the user

```

查看meeting子命令的帮助信息，用法：

1. agenda help meeting
2. agenda meeting --help

```

→ ~ agenda meeting --help
agenda meeting --help

1.To add Participator of the meeting:
instruction:    meeting addUser -p [Participator] -t [Title]

    [Participator] Participator's name
    [Title] meeting's name

(attention:If the Participator cannot attend during the time, add fail.)
(PS: There can be multiple -p flags at the same time, but each -p flag only aims at one participator.)

2. To remove a Participator of the meeting
instruction:    meeting deleteUser -p [Participator] -t [Title]

    [Participator] Participator's name
    [Title] meeting's name

3. To create a new meeting:
instruction:    meeting create -t [Title] -p [Participator] -s [StartTime] -e [EndTime]

    [Title] the Title of the meeting
    [Participator] the Participator of the meeting,the Participator can only attend one meeting during one meeting time
    [StartTime] the StartTime of the meeting
    [EndTime] the EndTime of the meeting

```

```

4. To cancel a meeting:

instruction:    meeting cancel -t [title]

               [Title] the Title of the meeting
               (PS: Only creator of this meeting can cancel it.)

5. To query meetings in the specific time for login user:

instruction:    meeting lookup -s [StartTime] -e [EndTime]

               [StartTime] the StartTime of the meeting
               [EndTime] the EndTime of the meeting

6. To quit a meeting:

instruction:    meeting exit -t [title]

               [Title] the Title of the meeting

               (attention: if there is no participators in this meeting,the meeting will be deleted)

7. To clear all meetings:

instruction:    clear

Usage:
agenda meeting [flags]

Flags:
-e, --end string      Help message for end time
-h, --help            help for meeting
-p, --participant stringArray  Help message for participant
-s, --start string    Help message for start time
-t, --title string    Help message for meeting title

```

同样查看用户命令下的其他子命令用法，如 `agenda help meeting addUser` 等。

```

➔ ~ agenda help meeting addUser
agenda help meeting addUser

Command : meeting addUser
Function: Add users to a meeting. It require all the users don't have meetings at the period this meeting held.
instruction: meeting addUser --title/-t [Title] --participant/-p [Participants]
Args :
    [Title] title of the meeting (unique)
    [Participants] Participants of the meetings
(PS: If multiple participants add at the same time, need multiple -p flag.
This : meeting addUser -p "wty" "wyx" is not available
But this : meeting addUser -p "wty" -p "wyx" is available)

➔ ~ agenda help meeting deleteUser
agenda help meeting deleteUser

Command : meeting deleteUser
Function: Delete user from the meeting. If this operation make the meeting empty, then delete the meeting.
instruction: meeting deleteUser --title/-t [Title] --participant/-p [Participants]
Args :
    [Title] title of the meeting (unique)
    [Participants] Participants of the meetings
(PS: If multiple participants delete at the same time, need multiple -p flag.
This : meeting deleteUser -p "wty" "wyx" is not available
But this : meeting deleteUser -p "wty" -p "wyx" is available)

```

→ ~ agenda help meeting create
agenda help meeting create

Command : meeting create
Function: Create a meeting.
instruction: meeting create --startTime/-s [StartTime] --endTime/-e [EndTime] --title/-t [Title] --participant/-p [Participants]
Args :
 [Title] title of the meeting (unique)
 [Participants] Participants of the meetings
 [StartTime] the StartTime of the meeting
 [EndTime] the EndTime of the meeting
(PS: If multiple participants delete at the same time, need multiple -p flag.
This : meeting create -p "wty" "wyx" is not available
But this : meeting create -p "wty" -p "wyx" is available)

→ ~ agenda help meeting cancel
agenda help meeting cancel

Command : meeting cancel
Function: User can cancel the meeting created by himself.
instruction: meeting cancel --title/-t [Title]
Args :
 [Title] title of the meeting (unique)

→ ~ agenda help meeting lookup
agenda help meeting lookup

Command : meeting lookup
Function: Login user can look up the meeting he attended in the specific period.
instruction: meeting create --startTime/-s [StartTime] --endTime/-e [EndTime]
Args :
 [StartTime] the StartTime of the meeting
 [EndTime] the EndTime of the meeting

→ ~ agenda help meeting exit
agenda help meeting exit

Command : meeting exit
Function: User can exit the meeting he attended.
instruction: meeting cancel --title/-t [Title]
Args :
 [Title] title of the meeting (unique)

→ ~ agenda help meeting clear
agenda help meeting clear

Command : meeting clear
Function: User can clear all the meetings he created.
instruction: meeting clear

5.3 user子命令

一、lookup

由于需要在运行agenda命令的目录下中保存user.txt和cache.txt，估进入goAgenda目录。

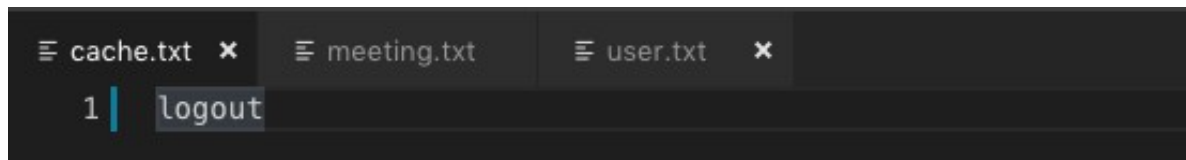
初始时在cache.txt保存着一个用户的登录信息，故可以直接使用lookup命令。


```

→ goAgenda git:(master) ✕ agenda user lookup
agenda user lookup
user lookup
Here is users' info:
dcy wtysos11@163.com 18028622384
wtysos11 wtysos11@163.com 18028622384
→ goAgenda git:(master) ✕ agenda user logout
agenda user logout
user logout
Logout success
→ goAgenda git:(master) ✕ agenda user lookup
agenda user lookup
user lookup
Please login first.
→ goAgenda git:(master) ✕ agenda user delete
agenda user delete
user delete
Please login first.

```

退出后cache.txt为logout状态，且只能使用register和login命令（见上图）。



二、login

```

→ goAgenda git:(master) ✕ agenda user login -u dcy
agenda user login -u dcy
user login
Must have a username and a password
→ goAgenda git:(master) ✕ agenda user login -p 123
agenda user login -p 123
user login
Must have a username and a password
→ goAgenda git:(master) ✕ agenda user login -u dcy -p 123
agenda user login -u dcy -p 123
user login
login failed
→ goAgenda git:(master) ✕ agenda user login -u dcy -p SKY__
agenda user login -u dcy -p SKY__
user login
Login success. Welcome!

```

三、register


```

→ goAgenda git:(master) ✖ agenda user register -u dcy
agenda user register -u dcy
register
Repeat Username
→ goAgenda git:(master) ✖ agenda user register -u testUser -p 123
agenda user register -u testUser -p 123
register
Must have an email
→ goAgenda git:(master) ✖ agenda user register -u testUser -p 123 -e "www"
agenda user register -u testUser -p 123 -e "www"
register
Must have a telephone
→ goAgenda git:(master) ✖ agenda user register -u testUser -p 123 -e "www" -t "111"
agenda user register -u testUser -p 123 -e "www" -t "111"
register
User register success

```

四、delete

```

→ goAgenda git:(master) ✖ agenda user lookup
agenda user lookup
user lookup
Here is users' info:
wtysos11 wtysos11@163.com 18028622384
testUser www 111
dcy wtysos11@163.com 18028622384
→ goAgenda git:(master) ✖ agenda user delete
agenda user delete
user delete
→ goAgenda git:(master) ✖ agenda user lookup
agenda user lookup
user lookup
Please login first.
→ goAgenda git:(master) ✖ agenda user login -u dcy -p SKY__
agenda user login -u dcy -p SKY__
user login
Login success. Welcome!
→ goAgenda git:(master) ✖ agenda user lookup
agenda user lookup
user lookup
Here is users' info:
wtysos11 wtysos11@163.com 18028622384
dcy wtysos11@163.com 18028622384

```

5.4 meeting子命令

一、lookup

```

→ goAgenda git:(master) ✖ agenda meeting lookup -s 1 -e 1
agenda meeting lookup -s 1 -e 1
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting lookup
Time format incorrect, don't have /
→ goAgenda git:(master) ✖ agenda meeting lookup -s "2019-11-12/11:11:11" -e "2020-11-10/11:11:11"
agenda meeting lookup -s "2019-11-12/11:11:11" -e "2020-11-10/11:11:11"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting lookup
Meeting lookup:

```

二、addUser

```

→ goAgenda git:(master) ✖ agenda meeting addUser
agenda meeting addUser
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
add user
Add user command needs a user list.
→ goAgenda git:(master) ✖ agenda meeting addUser -p 111 -t "111"
agenda meeting addUser -p 111 -t "111"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
add user
Invalid title.
→ goAgenda git:(master) ✖ agenda meeting addUser -p 111 -t "tts"
agenda meeting addUser -p 111 -t "tts"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
add user
Participants have illegal participant:111

```

```

→ goAgenda git:(master) ✖ agenda meeting addUser -p dcy -t "tts"
agenda meeting addUser -p dcy -t "tts"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
add user
Meeting add users success

```

三、deleteUser

```

→ goAgenda git:(master) ✖ agenda meeting deleteUser
agenda meeting deleteUser
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11 dcy
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
delete user
Add user command needs a user list.
→ goAgenda git:(master) ✖ agenda meeting deleteUser -p 111
agenda meeting deleteUser -p 111
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11 dcy
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
delete user
Add user command needs title
Participants have illegal participant:111
→ goAgenda git:(master) ✖ agenda meeting deleteUser -p dcy -t "111"
agenda meeting deleteUser -p dcy -t "111"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11 dcy
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
delete user
Meeting delete users failed.

```

```

→ goAgenda git:(master) ✖ agenda meeting deleteUser -p dcy -t "tts"
agenda meeting deleteUser -p dcy -t "tts"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11 dcy
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
delete user
Meeting delete users success

```

六、cancel

→ goAgenda git:(master) ✖ agenda meeting cancel
agenda meeting cancel
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting cancel
Meeting cancel command needs a title.
Meeting cancel failed
→ goAgenda git:(master) ✖ agenda meeting cancel -t "111"
agenda meeting cancel -t "111"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting cancel
Meeting cancel failed
→ goAgenda git:(master) ✖ agenda meeting cancel -t "test"
agenda meeting cancel -t "test"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting cancel
You can only cancel the meeting that create by yourself

→ goAgenda git:(master) ✖ agenda meeting cancel -t "ttse"
agenda meeting cancel -t "ttse"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
dcy 2019-11-12/11:11:11 2020-11-10/11:11:11 ttse wtysos11
meeting called
meeting cancel
Meeting cancel success
→ goAgenda git:(master) ✖ agenda meeting lookup
agenda meeting lookup
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2016-11-10/11:11:11 2017-11-10/11:11:11 tts wtysos11
meeting called
meeting lookup
Meeting lookup command needs a start time.

七、create

→ goAgenda git:(master) ✖ agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11
agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
meeting called
create
Meeting must have a title
→ goAgenda git:(master) ✖ agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11 -t "test"
agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11 -t "test"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
meeting called
create
Repeat title
→ goAgenda git:(master) ✖ agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11 -t "test_title"
agenda meeting create -s 2017-12-01/11:11:11 -e 2017-12-10/11:11:11 -t "test_title"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
meeting called
create
Meeting create success

八、exit

```
→ goAgenda git:(master) ✗ agenda meeting exit
agenda meeting exit
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test_title dcy
meeting called
meeting exit
Meeting Exit failed
→ goAgenda git:(master) ✗ agenda meeting exit -t "finish"
agenda meeting exit -t "finish"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test_title dcy
meeting called
meeting exit
Meeting Exit failed
→ goAgenda git:(master) ✗ agenda meeting exit -t "test"
agenda meeting exit -t "test"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11 dcy
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test_title dcy
meeting called
meeting exit
Meeting Exit success
```

```
→ goAgenda git:(master) ✗ agenda meeting exit -t "test_title"
agenda meeting exit -t "test_title"
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test_title dcy
meeting called
meeting exit
Meeting Exit success
```

```
→ goAgenda git:(master) ✗ agenda meeting lookup
agenda meeting lookup
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
meeting called
meeting lookup
Meeting lookup command needs a start time.
Time format incorrect, don't have /
```

九、clear

```
→ goAgenda git:(master) ✖ agenda meeting lookup
agenda meeting lookup
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test—title
meeting called
meeting lookup
Meeting lookup command needs a start time.
Time format incorrect, don't have /
→ goAgenda git:(master) ✖ agenda meeting clear
agenda meeting clear
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
dcy 2017-12-01/11:11:11 2017-12-10/11:11:11 test—title
meeting called
meeting clear
Meeting clear success
→ goAgenda git:(master) ✖ agenda meeting lookup
agenda meeting lookup
In reading Meeting file
Creator StartTime EndTime Title UserList
wtysos11 2018-10-29/11:11:11 2019-11-11/11:11:11 test wtysos11
wtysos11 1018-10-29/11:11:11 1019-11-11/11:11:11 finish wtysos11
meeting called
meeting lookup
Meeting lookup command needs a start time.
Time format incorrect, don't have /
```

6. 项目组内成员与分工

姓名	学号	工作
王迎旭	16340226	实验报告与测试程序
吴天扬	16340240	整体框架的搭建、user与meeting功能的完善
王泽浩	16340232	debug与优化提示信息
吴聪	16340237	完善日志功能、完善程序接口

7. 源代码

[Github](#)

8. 参考资料

- [服务计算——goAgenda](#)
- [Cobra官方文档](#)
- [Json与Go](#)

