

GOLANG 作品

# --會員註冊登入網站--

報告者:陳泓睿

# 目錄

- 簡介
- 流程圖
- 工具
- 作品展示

# 簡介

使用go語言結合html、css、javascript及mysql，製作會員登入網站，並使用Heroku部署，讓使用者透過api在web上操作使用。

# 製作流程圖

go語言撰寫會員註冊登入  
系統後端程式



html、css、javascript撰寫  
會員註冊登入系統網頁程式

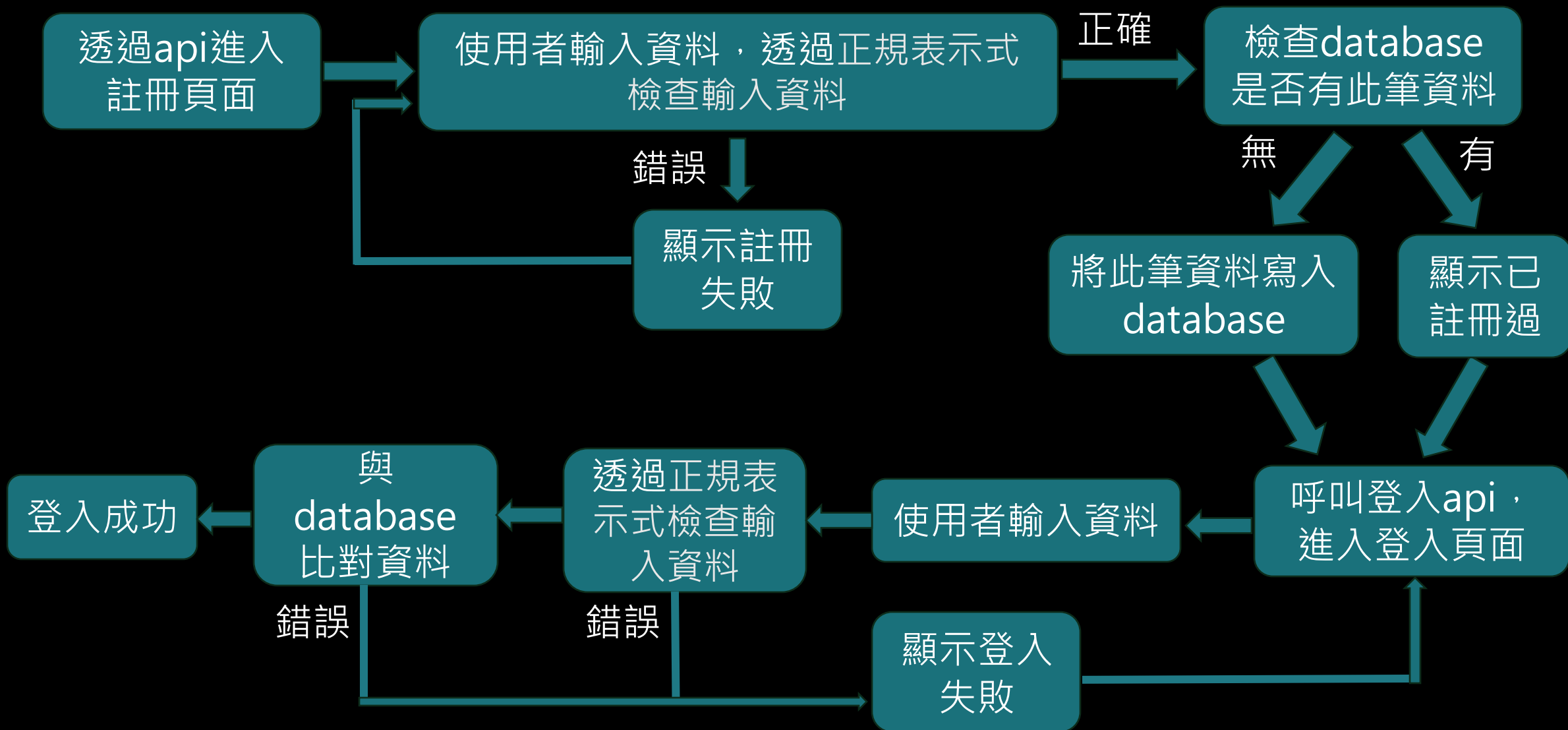


透過api讓使用者在web上  
操作使用



mysql存放會員資料  
(account、password、  
email、phone)

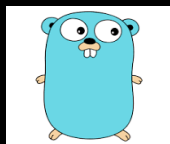
# 邏輯流程圖



# 使用工具



使用vscode撰寫code及程式執行測試



使用go語言撰寫會員註冊登入系統後端程式



使用html、css、javascript撰寫會員註冊登入系統網頁程式



使用MySQL存放會員資料(account、password、email、phone)



使用Herokur將代碼部署至雲端

# 作品展示

```
1 package main
2 import {
3     "database/sql"
4     "fmt"
5     "log"
6     "net/http"
7     "html/template"
8 }
9
10 // "github.com/go-sql-driver/mysql"
11
12 // const cardNum int = 13
13
14 var db *sql.DB
15
16 // 開機後 - init() 初始化 - 時間點比 main() 更早 -
17 func init() {
18     dbConnect, err := sql.Open(
19         "mysql",
20         "root:root@tcp(127.0.0.1:3306)/golang",
21     )
22     if err != nil {
23         log.Fatalln(err)
24     }
25     err = dbConnect.Ping()
26     if err != nil {
27         log.Fatalln(err)
28     }
29 }
30
31 db = dbConnect // 用全域變數
32
33 db.SetMaxOpenConns(10) // 可設置最大DB連線數 - 設<=0則無上限 (連線分成 in-use正在執行任務 及 idle執行完成後的閒置 兩種)
34 db.SetMaxIdleConns(10) // 設置最大idle閒置連線數
35 }
```

```
1 // 帳號密碼姓名Email
2
3 func login(w http.ResponseWriter, r *http.Request) {
4     if r.Method == "GET" {
5         // t1, err := template.ParseFiles("view/login2.html")
6         t1, err := template.ParseFiles("login.html")
7         if err != nil {
8             panic(err)
9         }
10        t1.Execute(w, "")
11    } else {
12        r.ParseForm()
13
14        var account string
15        var password string
16        for k, v := range r.Form {
17            if k == "account" {
18                account = v[0]
19            } else if k == "password" {
20                password = v[0]
21            }
22        }
23
24        rows, err := db.Query("SELECT COUNT(*) FROM 'web_test'.user WHERE 'account'=? AND 'password'=?", account, password)
25        var tCount int
26        for rows.Next() { // rows.Next() 前往下一個項目 - 如果成功 (還有下一項的話) 返回True - 失敗 (沒有下一項可讀) 則返回False
27            err = rows.Scan(&tCount) // 掃描後存進變數中
28            if err != nil {
29                log.Fatalln(err)
30            }
31            fmt.Printf("%q %d\n", tCount) // %q:quoted 用引號包起字串
32        }
33        if tCount == 1 {
34            fmt.Fprint(w, "登入成功")
35        } else {
36            fmt.Fprint(w, "登入失敗")
37        }
38        defer rows.Close()
39    }
40 }
```

```
1 func main() {
2
3     createDb("web_test")
4     _, err := db.Exec("CREATE TABLE IF NOT EXISTS 'web_test'.user('account' VARCHAR (20), 'password' VARCHAR (20), 'email' VARCHAR (100), 'phone' VARCHAR (20))")
5     if err != nil {
6         log.Fatalln(err)
7     }
8
9     http.HandleFunc("/login", login)
10    http.HandleFunc("/signup", Signup)
11
12    err = http.ListenAndServe("localhost:8080", nil)
13
14    if err != nil {
15        log.Fatal("ListenAndServe: ", err)
16    }
17 }
```

```
1 func Signup(w http.ResponseWriter, r *http.Request) {
2     if r.Method == "GET" {
3         // t1, err := template.ParseFiles("view/signup.html")
4         t1, err := template.ParseFiles("signup.html")
5         if err != nil {
6             panic(err)
7         }
8         t1.Execute(w, "")
9     } else {
10        r.ParseForm()
11
12        fmt.Println(r.Form)
13
14        var account string
15        var password string
16        var email string
17        var phone string
18        for k, v := range r.Form {
19            if k == "account" {
20                account = v[0]
21            } else if k == "password" {
22                password = v[0]
23            } else if k == "email" {
24                email = v[0]
25            } else if k == "phone" {
26                phone = v[0]
27            }
28        }
29        rows, err := db.Query("SELECT COUNT(*) FROM 'web_test'.user WHERE 'account'=? AND 'password'=?;", account, password)
30        var tCount int
31        for rows.Next() { // rows.Next() 前往下一個項目 - 如果成功 (還有下一項的話) 返回True - 失敗 (沒有下一項可讀) 則返回False
32            err = rows.Scan(&tCount) // 掃描後存進變數中
33            if err != nil {
34                log.Fatalln(err)
35            }
36            fmt.Printf("%q %d\n", tCount) // %q:quoted 用引號包起字串
37        }
38        if tCount == 0 {
39            _, err := db.Exec("INSERT INTO 'web_test'.user ('account' , 'password' , 'email' , 'phone') VALUES (?, ?, ?, ?)", account, password, email, phone)
40            if err != nil {
41                log.Fatalln("INSERT:", err)
42            } else {
43                http.Redirect(w, r, "/login", http.StatusSeeOther)
44            }
45        } else {
46            fmt.Fprint(w, "該帳號已註冊")
47        }
48    }
49 }
```

```
1 func createDb(dbName string) {
2     _, err := db.Exec("CREATE DATABASE IF NOT EXISTS " + dbName + ";")
3     if err != nil {
4         log.Fatalln(err)
5     }
6     fmt.Print(err)
7 }
8 }
```

# 作品展示

主機: 127.0.0.1

資料庫: login

資料表: member

資料

查詢\*

1 INSERT INTO `login`.`member` (`account`,`password`,`email`,`phone`) VALUES ('aaa','123','aaa@gmail.com',

2 SELECT `account`,`password`,`email`,`phone` FROM `login`.`member` WHERE `account`='aaa' AND `password`='12

3

4 INSERT INTO `login`.`member` (`account`,`password`,`email`,`phone`) VALUES ('bbb','456','bbb@gmail.com',

5 SELECT `account`,`password`,`email`,`phone` FROM `login`.`member` WHERE `account`='bbb' AND `password`='45

6

7 INSERT INTO `login`.`member` (`account`,`password`,`email`,`phone`) VALUES ('ccc','789','ccc@gmail.com',

8 SELECT `account`,`password`,`email`,`phone` FROM `login`.`member` WHERE `account`='ccc' AND `password`='78

9

10 SELECT NOW();

主機: 127.0.0.1

資料庫: login

資料表: member

資料

查詢\*

login.member: 3 條資料 (exact)

#	account	password	email	phone
1	aaa	123	aaa@gmail.com	0123456789
2	bbb	456	bbb@gmail.com	012345679
3	ccc	789	ccc@gmail.com	0123456789



# 作品展示

