

Go语言基础之文件操作→B站视频链接

Go语言标准库之strconv

Go语言标准库之template

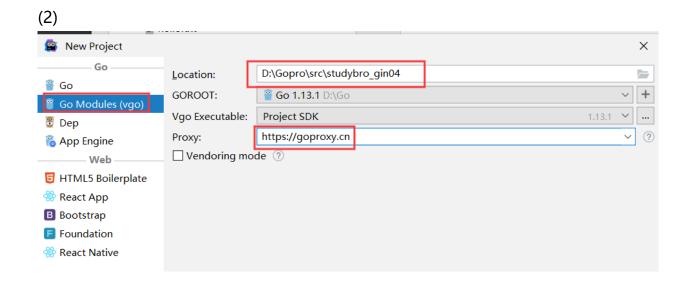
Go语言标准库之net/http

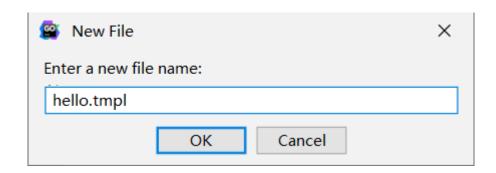
Go语言标准库之context

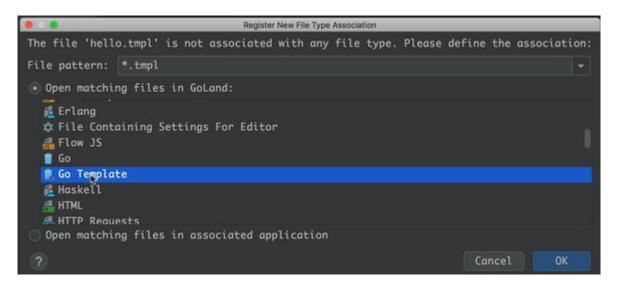
持续更新中...

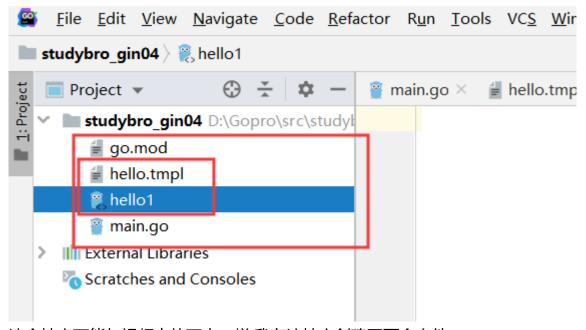
框架

go语言内置模板的基本使用.









这个地方可能与视频中的不太一样,我在这地方创建了两个文件.

(3)

目录结构:

```
studybro gin04 D:\Goprc 22
                            23
    🗐 go.mod
                            24
      hello.tmpl
                            25
    % hello1
                            26
    🆀 main.go
                            27
    📆 studybro_gin04.exe
                            28
IIII External Libraries
                            29
Scratches and Consoles
                            30
```

main.go

```
1 package main
3 import (
  "fmt"
4
   "net/http"
5
   "html/template"
7 )
  //遇事不决,先写注释,受益终身的一句话
9
10
   func sayHello(w http.ResponseWriter,r *http.Request){
11
    //2.解析模板,遇事不决,先写注释
12
   //func ParseFiles(filenames ...string) (*Template, error)
13
   t,err :=template.ParseFiles("./hello1")
14
   if err!=nil{
15
   fmt.Println("Parse template failed,err:%v",err)
16
    return
17
   }
18
   //3. 渲染模板
19
   //err=t.Execute(w,"小冬瓜")
20
    name:="小西瓜"
21
   err=t.Execute(w,name)
22
   if err!=nil{
23
   fmt.Println("render template failed,err:%v",err)
24
    return
25
    }
26
27
   }
28
29 func main() {
```

```
http.HandleFunc("/",sayHello)

rer:=http.ListenAndServe(":9000",nil)

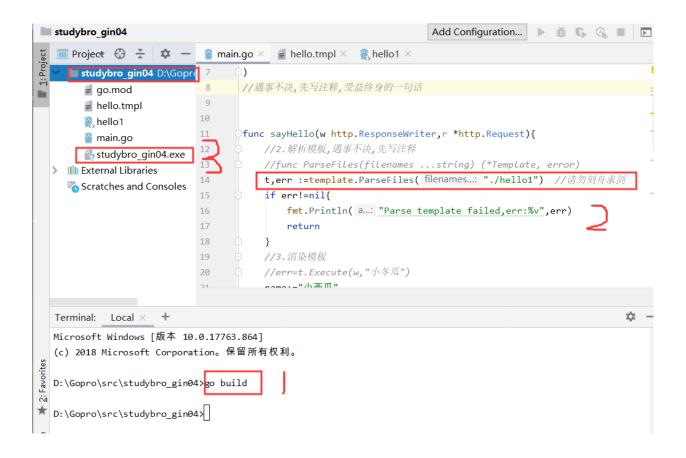
fmt.Println("Http server start failed,err:%v",err)

return

}
```

hello

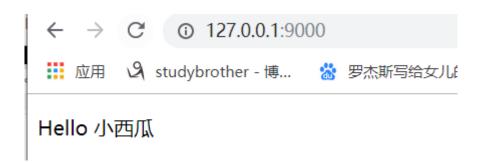
```
1 <!DOCTYPE html>
2 <html lang="zh-CN">
3 <head>
4 {{/* <meta charset="UTF-8">*/}}
5 {{/* <meta name="viewport" content="width=device-width, initial-scale=1.
0">*/}}
6 {{/* <meta http-equiv="X-UA-Compatible" content="ie=edge">*/}}
7 <title>Hello</title>
8 </head>
9 <body>
10 Hello {{.}}
11 </body>
12 </html>
```



D:\Gopro\src\studybro_gin04>studybro_gin04.exe

此时,我们再次执行生成的这个文件.在上图中会夯住,我们需要在浏览器中输入下面的信息 127.0.0.1:9000,得到下面对应的内容

(4)



相当于是,我们将name替换html文件中的{{.}}

本质上还是字符串替换