

# 项目chatroom

---

## 学习了Tcp协议的调用

### 1: 流程

```
服务器
ln,err:=net.Listen("tcp",":8088")
if err!=nil{
    fmt.Printf("Listen failed: %v",err)
    os.Exit(-1)
}
defer ln.Close()
conn,err:=ln.Accept()
if err!=nil{
    fmt.Printf("Accept failed: %v",err)
    os.Exit(-1)
}
defer conn.Close()
客户端
conn,err:=Dial("tcp",":8088")
if err!=nil{
    fmt.Printf("Dial failed: %v",err)
}
```

## Mysql的使用

### open函数的理解

```
sql.Open("mysql","use:password@/dbname") 第一个是一个驱动引擎，第二个是你的账户和密码
以及你要连接的数据库
Open函数 does not return an error.所以我们需要使用db.Ping()的返回值去尝试是不是能成
功。例如
err = db.Ping()
if err != nil {
    panic(err.Error()) // proper error handling instead of panic in your app
}
```

### Prepare()函数的理解

Prepare()函数是为了准备一个声明，也就是说我将要干什么，但是并没有真正的执行它。为后面的多行的插入或者查找提供了可能 形式是stmtint,err:=db.Prepare("insert into tablename values(?,?)") Exec()函数执行一行的插入或者查找但是不返回任何row(Exec executes a query without returning any rows.) \_err:=stmtint.Exec(1,2) 使用单行查找的话也可以使用stmtOut.QueryRow(2).Scan(&squareNum).?'是个占位符

### 多行数据的查找

```
-- Query()函数可以去查找多行的数据(Query executes a query that returns rows, typically a SELECT.) -- rows,
err:=db.Query("select *from table") -- var str string -- for row.Next(){ -- var name string -- var sex string --
rows.Scan(&name,&sex) -- str +=name -- }
```