

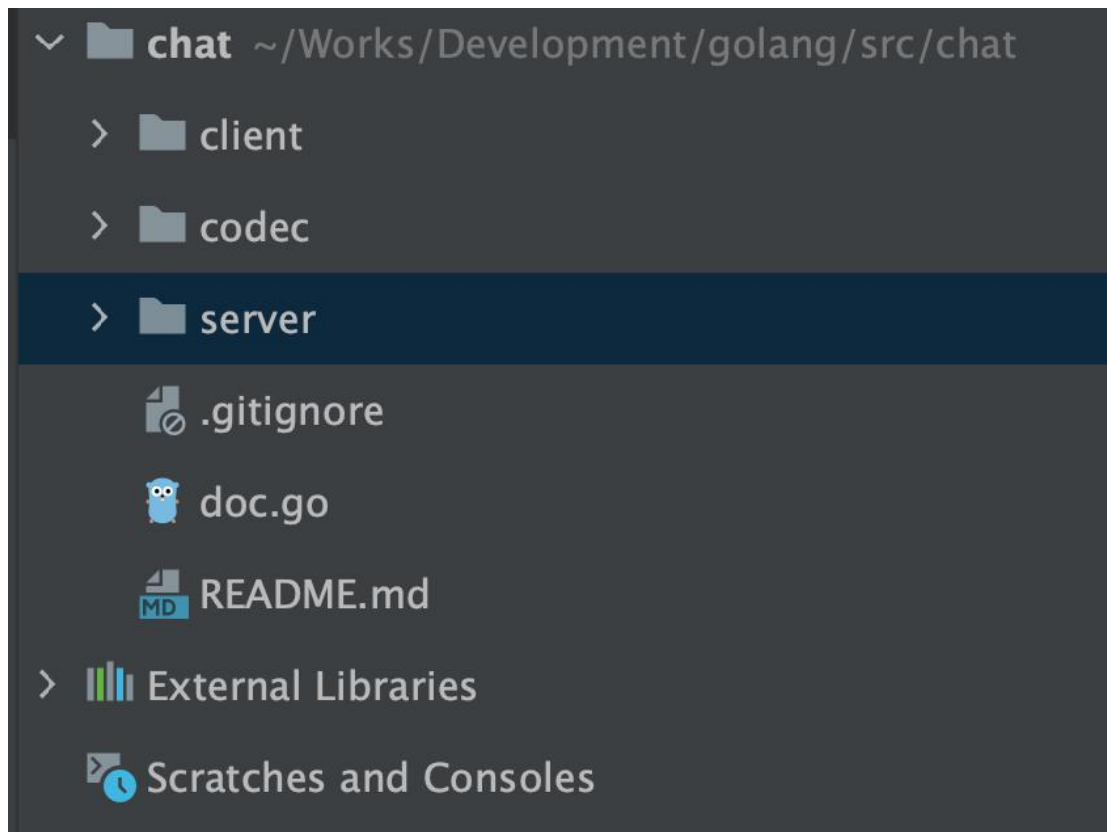
# chat 项目说明

## 一、项目简介

chat 项目是一个基于 golang 实现的玩家聊天室应用。

源码地址: <https://github.com/huangyuCN/chat>

为了方便测试, chat 项目将客户端和服务端代码放在了一个工程中。



如图所示: client 保存的是客户端源码; server 保存的是服务器源码。

## 二、服务器逻辑架构

服务器的主要功能包括:

### 1. TCP 链接管理

本项目使用 TCP 长连接实现消息收发, 服务器发送到客户端消息使用 JSON 格式, 客户端发送到服务器消息使用字符串。

server/connManager 管理一个 tcp 链接对象和一个 user 对象, 他们一一对应。

### 2. 用户管理

用户管理包含两个对象, server/userManager 和 server/user。

server/userManager 用来管理服务器上的所有玩家。保证玩家注册时名字唯一。

server/user 是玩家对象, 它管理玩家的基本个人信息已经玩家消息的收发和路由。

### 3. 聊天室管理

聊天室管理包含两个对象, server/roomManager 和 server/room。

server/roomManager 管理聊天室的创建和玩家加入聊天室。保证聊天室名字唯一。

server/room 是聊天室对象, 聊天室维护了创建者信息; 所有在该聊天室的玩家信息; 历史消息保存; 群消息收发等。

#### 4. 消息编码

`server/message` 对象是玩家在群聊中发送的所有消息的封装对象，每条消息都包含发送者、发送时间、内容、是否是 GM 消息、GM 消息参数。消息发送到客户端时会打包为 JSON 格式。

#### 5. 敏感词过滤

`server/filter` 是基于 trie 树数据结构的 DAF 算法的 golang 实现。用来做敏感词匹配和替换。

#### 6. 单词频率统计

`server/popular` 用来管理所有玩家聊天中出现的单词的频率。最多只记录最近 60 秒内的单词出现次数。查询的时候可以传入时间参数来缩小查询范围。如果查询的时间段内没有聊天消息，则可能返回空。

### 三、使用方法

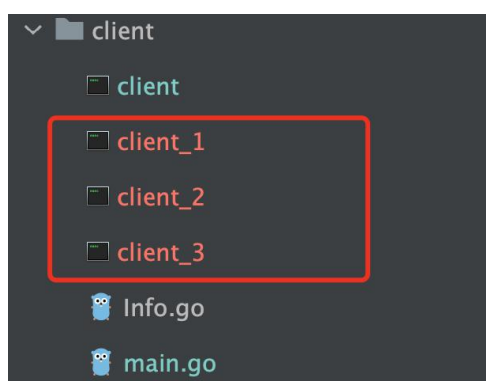
本节内容是基于 macOS Big Sur 的使用说明，不同操作系统稍有不同。

#### 1. 启动服务器

可以使用 `server/main.go` 进行源码启动，也可以打包后再启动。

#### 2. 客户端启动

因为需要测试多用户的聊天室功能，因此客户端可以打包成多个应用后使用在“终端”启动。



如图所示，打包了三个不同名字的客户端。

```
→ client git:(main) x ./client_1
tcp connected...

-----From:system Time:2021-07-31 12:15:27 PM -----
The GM Commands list:
/help = show all commands
/register [name] = register a new user with name
/rooms = show all chat rooms
/createRoom [name] = create a chat room with name
/leaveRoom = leave the chat room
/joinRoom [name] = join the chat room by name
/closeRoom = close the room, if the room create by yourself
/users = show all users registered
/popular [n] = show the popular word in recent n seconds, n less than 61
/stats [username] = show the online time of user whose name is username
/quit = disconnect with server
|
```

客户端启动后，会看到所有的 GM 命令提示。

需要说明的是，在进行聊天之前，需要先注册。使用 `/register [name]` 命令进行注册，例如 `/register test` 表示注册一个名字为 `test` 的玩家。

玩家可以自己创建聊天室，也可以加入别人创建的聊天室。使用 `/createRoom [name]` 命令创建自己的聊天室，例如 `/createRoom test` 表示创建一个名字为 `test` 的聊天室。

以下是所有命令的使用说明：

命令	说明	示例
<code>/help</code>	查询所有指令	<code>/help</code>
<code>/register [name]</code>	玩家注册	<code>/register test</code>
<code>/rooms</code>	查询当前所有的聊天室	<code>/rooms</code>
<code>/createRoom [name]</code>	创建聊天室	<code>/createRoom test</code>
<code>/leaveRoom</code>	离开当前聊天室	<code>/leaveRoom</code>
<code>/joinRoom [name]</code>	加入聊天室	<code>/joinRoom test</code>
<code>/closeRoom</code>	创建者关闭聊天室	<code>/closeRoom</code>
<code>/users</code>	服务器上在线的所有玩家	<code>/users</code>
<code>/popular [n]</code>	查询 <code>n</code> 秒内出现的最高频词	<code>/popular 60</code>
<code>/stats [name]</code>	查询玩家的在线时间	<code>/stats test</code>
<code>/quit</code>	断开与服务器的链接	<code>/quit</code>

### 3. 运行效果

```
→ client git:(main) x ./client_1
tcp connected...

-----From:system Time:2021-07-31 01:21:40 PM -----
The GM Commands list:
/help = show all commands
/register [name] = register a new user with name
/rooms = show all chat rooms
/createRoom [name] = create a chat room with name
/leaveRoom = leave the chat room
/joinRoom [name] = join the chat room by name
/closeRoom = close the room, if the room create by yourself
/users = show all users registered
/popular [n] = show the popular word in recent n seconds, n less than 61
/stats [username] = show the online time of user whose name is username
/quit = disconnect with server

/register c1

-----From:system Time:2021-07-31 01:22:04 PM -----

/createRoom c2

-----From:system Time:2021-07-31 01:22:33 PM -----
chat room create success
/rooms

-----From:system Time:2021-07-31 01:22:37 PM -----
c2
haha

-----From:c1 Time:2021-07-31 01:22:48 PM -----
haha
```

-----From:system Time:2021-07-31 01:23:33 PM -----  
c2 join in the chat

-----From:c2 Time:2021-07-31 01:23:36 PM -----  
haha bibi

-----From:system Time:2021-07-31 01:23:43 PM -----  
c3 join in the chat

-----From:c3 Time:2021-07-31 01:23:47 PM -----  
haha cici  
fuckyou

-----From:c1 Time:2021-07-31 01:24:01 PM -----  
\*\*\*you  
/popular 60

-----From:system Time:2021-07-31 01:24:10 PM -----  
haha  
/stats c2

-----From:system Time:2021-07-31 01:24:14 PM -----  
00d 00h 01m 59s  
|