Light Chat

IEMS5722 Project ——Group 4

成员

姓名	学号	邮箱
林佑光	1155169171	1155169171@link.cuhk.edu.hk
贺亦欣	1155164941	1155164941@link.cuhk.edu.hk

个人贡献

林佑光

日期	贡献
2021.12.20	熟悉页面样式代码
2021.12.21	画流程图
2021.12.22	建完 mysql 数据库,造完假数据
2021.12.23	完成本地 MongoDB+SpringBoot（不过没做异常处理，就是正常自动返回，400 就客户端错误，500 就服务端错误，目前实现一个接口，其他接口照搬即可）

2021.12.24	研究 websocket

贺亦欣

日期	贡献
2021.12.20	
2021.12.21	
2021.12.22	研究 mybatis 连接数据库 shiro 安全认证
2021.12.23	
2021.12.24	实现 jwt 认证和登录，整合项目

待解决问题

1、@Repository 的作用

2、MongoDB 没专门做异常处理（YG）

3、Mysql 的 springboot 后端

任务表

Date	Task	
21	YG：画流程图， YX：研究登陆	
22	早上：YG，YX：讨论流程图，确认流程图	下午：YG：建数据库，建造假数据。

	和数据库	YX：安全登陆
23	YG：在本地跑后台，并连接 mongodb 数据库与 mysql 数据库	
24	YG：，暂时先实现显示群聊天（暂时先固定，不创建群聊），私人聊天。 YX：先在安卓里对接登陆功能，合并 MongoDB 代码部署到服务器	
25		
25	咱们就得在 25 号那天至少实现登陆和添加朋友一对一功能	

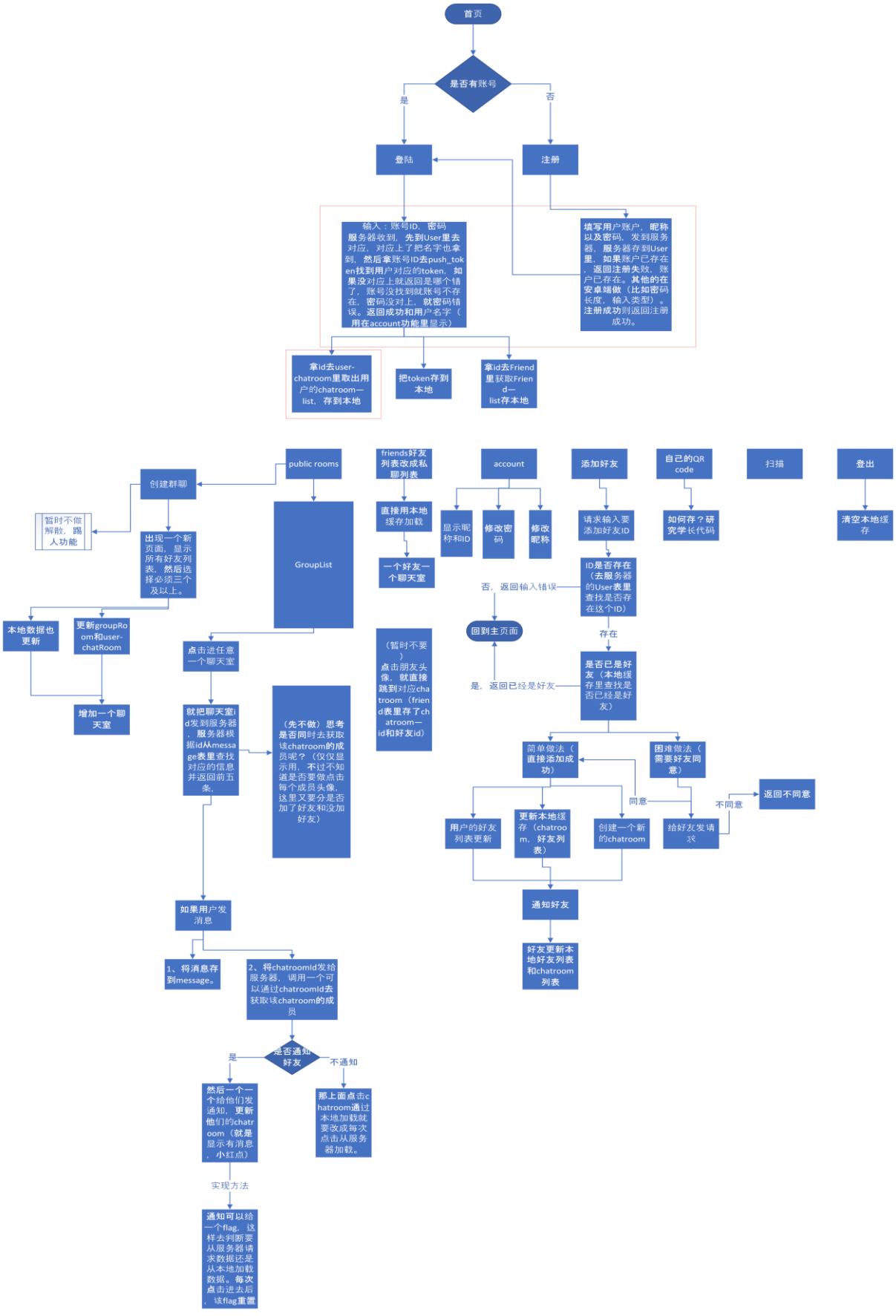
会议记录

Date	Content
12.21 pm 7:00	1、朋友也存 mongodb 2、添加一个好友 朋友列表里加一（自己，[name：好友，chatroom—id:1]）（好友，[name：我，chatroom—id:1] 3、聊天室列表里加一 （chatroom：1， GroupMemember：[我，你]) 4、朋友列表 User id: xxxx Friend_List: { {name:xx,chatroom:xx},{name:xx,chatroom:xx},{name:xx,chatroom:xx} } 5、关系表 用户-聊天室表、用户-聊天室表： user：用户名，单人聊天室列表：【】， 多人聊天室列表：【】 6、我发给你消息 -》数据库表更新 -〉

	判断你们两是否是第一次聊天 -》如果是-〉通过你的 id 找到你的 token -》 通过 token 给你设备传递信息 -〉你的设备就会收到通知 -》 收到通知就在会议列表里增加一个列表 -〉 你点击那个聊天室进去 -》 去服务器加载信息 -〉 显示 -〉如果不是（之前有过聊天，已经有这个聊天室） -》 通过 token 给你设备传递信息 -〉你的设备就会收到通知 -》点通知后进到聊天室才更新 7、本地：chatroom—id，通过这个来判断两个人是否创建过聊天室 8、跟 token 一样，第一次登陆的时候搞一份存在本地 9、退出的时候 就 情空 本地数组 10、然后 添加一个朋友，同时更新本地数据库 11、FCM -》 token （sdk 产生的），不是根据设备的 token 来发消息，咱们是 user 的 token（咱们自己产生的）socket 服务器和客户端一直保持通信 两个人之间通信 socket
12.22 pm 1:52	1、流程图画完后，得来确定后端需要哪些函数。 2、不做群里点击进去查看成员以及点击进入私聊 3、改成群聊，私聊两个功能 5、 account 里改名字和改密码 6、 群聊创建后在右上角加一个+号，点击拉人进去，必须拉三个人以上。 7、 token 的使用过程，有时间限制，一段时间断开重新登陆，如果有效，即使关掉重启也能直接登陆进去。
12.22 pm 20:00	1、 更新朋友列表问题 :当发起添加朋友的一方 ,发起请求后，目前暂时不考虑对方同不同意，直接当作对方同意，然后服务器创建一个 friendChatRoom。服务器返回 ChatRoom Id 后，同时更新 Friend 表里的添加方和被添加方的信息。

	2、把 chatroom 表格拆成 groupChatroom 和 friendChatroom，因为我们分成两个功能。
12.22 pm 21:00	1、删除 friendChatroom 里 friendChatName，不需要，在前端将其与朋友的名字合并当作名字就可以.
12.22 pm 22:00	1、由于 chatroom 拆成两个，所以 Message 要增加一个 type 来区分两个 chatroom，使用枚举类型，“group”，“friend”
12.22 am 12:00	1、将 user id 从 int 改成 string。

项目流程图


首页
是否有账号
是
否
登陆
注册
输入：账号ID，密码
服务器收到，先到User里去对应，对应上了把名字也拿到，然后拿账号ID去push_token找到用户对应的token，如果没对应上就返回是哪个错了，账号没找到就账号不存在，密码没对上，就密码错误。返回成功和用户名字（用在account功能里显示）
填写用户账户，昵称以及密码，发到服务器，服务器存到User里，如果账户已存在，返回注册失败，账户已存在。其他的在安卓端做（比如密码长度，输入类型）。注册成功则返回注册成功。
拿id去user-chatroom里取出用户的chatroom—list，存到本地
把token存到本地
拿id去Friend里获取Friend—list存本地
public rooms
friends好友列表改成私聊列表
account
添加好友
自己的QR code
扫描
登出
创建群聊
暂时不做解散，踢人功能
出现一个新页面，显示所有好友列表，然后选择必须三个及以上。
GroupList
直接用本地缓存加载
一个好友一个聊天室
显示昵称和ID
修改密码
修改昵称
请求输入要添加好友ID
如何存？研究学长代码
清空本地缓存
否，返回输入错误
ID是否存在（去服务器的User表里查找是否存在这个ID）
回到主页面
存在
是，返回已经是好友
是否已是好友（本地缓存里查找是否已经是好友）
本地数据也更新
更新groupRoom和user-chatRoom
增加一个聊天室
点击进任意一个聊天室
就把聊天室id发到服务器，服务器根据id从message表里查找对应的信息并返回前五条。
（先不做）思考是否同时去获取该chatroom的成员呢？（仅仅显示用，不过不知道是否要做点击每个成员头像，这里又要分是否加了好友和没加好友）
（暂时不要）
点击朋友头像，就直接跳到对应chatroom（friend表里存了chatroom—id和好友id）
简单做法（直接添加成功）
困难做法（需要好友同意）
同意
不同意
返回不同意
用户的好友列表更新
更新本地缓存（chatroom，好友列表）
创建一个新的chatroom
给好友发请求
通知好友
好友更新本地好友列表和chatroom列表
如果用户发消息
1、将消息存到message。
2、将chatroomId发给服务器，调用一个可以通过chatroomId去获取该chatroom的成员
是否通知好友
是
不通知
然后一个一个给他们发通知，更新他们的chatroom（就是显示有消息，小红点）
那上面点击chatroom通过本地加载就要改成每次点击从服务器加载。
实现方法
通知可以给一个flag，这样去判断要从服务器请求数据还是从本地加载数据。每次点击进去后，该flag重置

功能定义

1 个人聊天室

1.1 获取聊天室 ID

输入：userId 输出：用户好友列表

2 多人聊天室

2.1 获取聊天室 ID

输入：userId 输出：用户参与的所有群天室

接口文档

FCM

1、存储 Token

Method: POST	**URL:** /
Parameters:	
RequestBody:	
Format:	
Response:	
Description: 用户注册设备，然后把 token 存数据库	

2、群组里发消息并且通知群里所有人

3、私人发消息只通知朋友。

MYSQL

1、login

Method: POST	**URL:** /login
Parameters:	
RequestBody: {"id":id,"password":password}	
Format: JSON id--String password--String	

Response:
□□□□
{
"msg":"login succeed",
"code":1001,
"jwt_token":token
}
错误返回
查看具体的 http 状态码和错误信息
http 状态码 400
返回体 JSON
{
"code":1002
"msg":具体的错误信息
}
1002 表示用户名或密码错误
http 状态码 500
服务器内部错误
返回体为 paintext 查看具体的错误信息	

Description: 用户登录成功，刷新并返回 jwt_token	

2、register

Method: POST	**URL:**/register
Parameters: {"id":id,"username":username,"password":password}	
Format: String	
Response: □□□□ {	

"code",1001， "msg","register succeed"， "jwt_token",token } 异常返回 http status code 400 code 1002 用户已存在 http status code 500 服务器内部错误
Description: 用户注册成功，返回 jwt_token

3、其它请求

若返回 code 1003，说明 token 已过期/无效，需要用户重新登陆

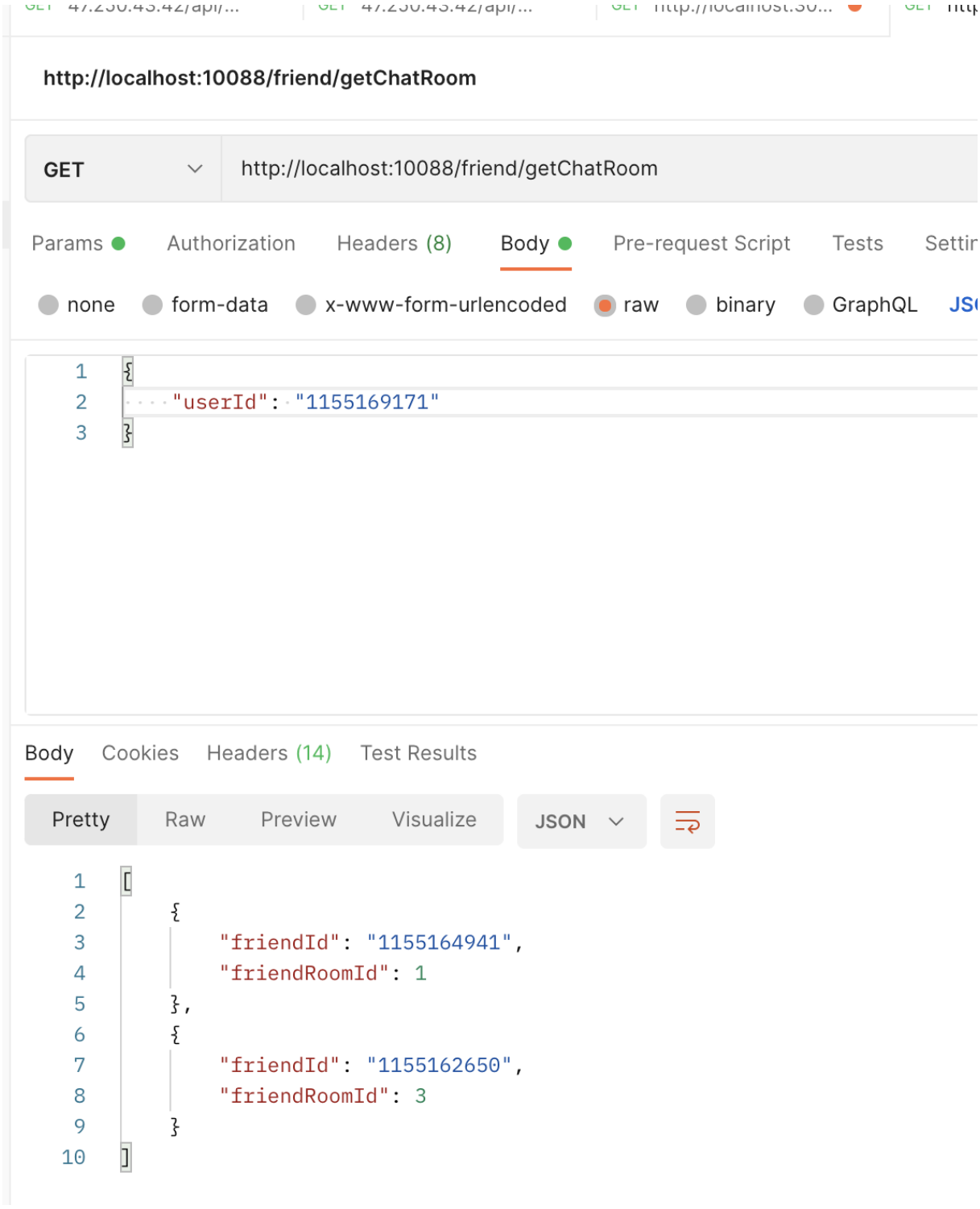
MongoDB

1、friend

Method:	**URL:**/friend/getChatRoom
Parameters: String userId	
Format: friendId: string , friendRoomId: int	

Response:
暂时只做了正确返回
{
{ friendId: "1155169171" , friendRoomId: 1 }
}
正正正正正正
400:正正正正正
500:正正正正正	

Description: 通过给指定的 userId 从 Friend 里获取该用户所有的好友及其对应的聊天室。	



2、friendChatRoom

Method: GET	**URL:**
Parameters:	
Format:	

Response:
Description:

2、groupChatRoom

Method: GET	**URL:** /
Parameters:	
Format:	

Response:
{
 	
}	

Description:	

3、userChatRoom

Method: GET	**URL:** /
Parameters:	
Format:	

Response:
{
 	
}	

Description:	

数据库

MongoDB：

1、Friend

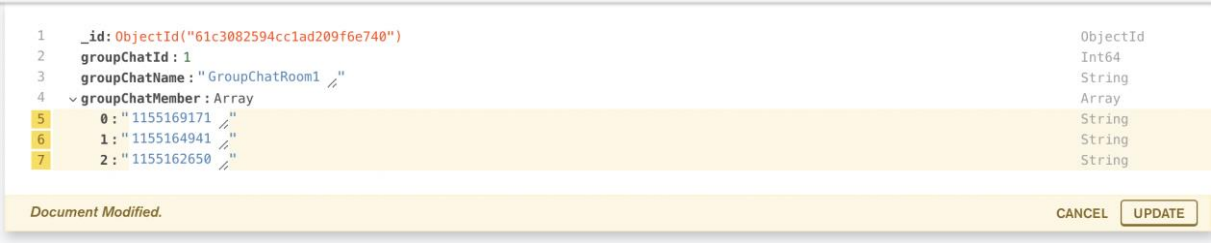
userId:	friendList
string	[{ friendId: string , friendRoomId: int }]

2、Chatroom

拆成 -》GroupList 、 FriendList - id 都由服务器产生

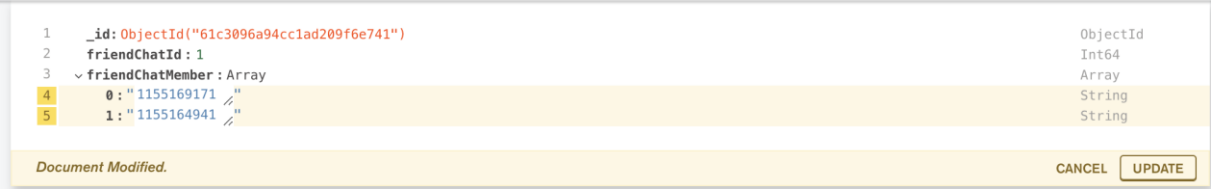
2.1 groupChatRoom

groupChatId	groupChatName	groupChatMember
int	string	[member_id1,member_id2,member_id3]string



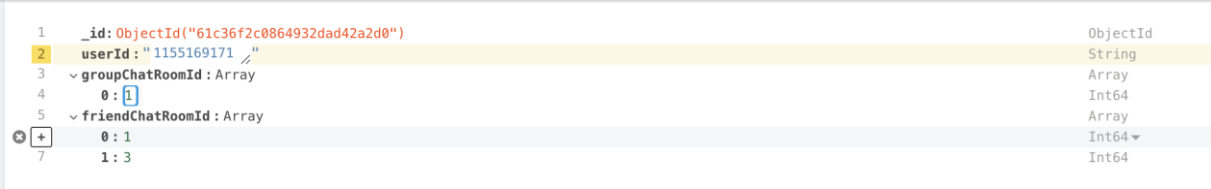
2.2 friendChatRoom

friendChatId	friendChatMember
int	[member_id1 , member_id2] string



3、userChatRoom

userId	groupChatRoomId	friendChatRoomId
int	[1 , 2 , 3 , etc]	[1,2,3]



MYSQL :

Drop table "table name;

1、user

id	name	password

String(10)	String(20)	String(20)

```
CREATE TABLE IF NOT EXISTS user( id VARCHAR(10),  name VARCHAR(20) NOT NULL,  password VARCHAR(20) NOT NULL, PRIMARY KEY ( id)) DEFAULT CHARSET=utf8mb4;
```

2、messages

id	type	chatroom_id	user_id	name（发送者）	message	message_time
Integer	ENUM	String(200)	String(20)	String(20)	String(200)	TImestamp

不需要接受者

```
CREATE TABLE messages ( id INT NOT NULL AUTO_INCREMENT,  type ENUM('group', 'friend'), chatroom_id VARCHAR(200) NOT NULL, user_id VARCHAR(20) NOT NULL, name VARCHAR(20) NOT NULL, message VARCHAR(200) NOT NULL, message_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (id) ) DEFAULT CHARSET = utf8;
```

3、pushToken

id	user_id（User表里的id）	token
int	string(20)	string(200)

```
CREATE TABLE IF NOT EXISTS pushToken( id INT UNSIGNED AUTO_INCREMENT,  user_id VARCHAR(20)  NOT NULL,  token VARCHAR(200) NOT NULL, PRIMARY KEY ( id)) DEFAULT CHARSET=utf8;
```

LocalStorage：

1、userToken: String

2、userChatroomList: []

3、userFriendList（用 List 存）：

```
{
    {friendId: xxx , chatroomId:xxx}
}
```

数据库数据

MongoDB

1、Friend

userId:	friendList
1155169171	[{friendId: 1155164941 , friendRoomId: 1}, {friendId: "1155162650", friendRoomId
1155164941	[{friendId: 1155169171 , friendRoomId: 1}, {friendId: "1155162650", friendRoomId
1155162650	[{friendId: 1155164941 , friendRoomId: 2}, {friendId: "1155169171", friendRoomId:

数据下载：https://github.com/Liny777/UserMongoDB_Altas/blob/master/Friend.json

2、groupChatroom

groupChatId	groupChatName	groupChatMember
1	GroupChatRoom1	["1155169171","1155164941","1155

数据下载：https://github.com/Liny777/UserMongoDB_Altas/blob/master/groupChatRoom.json

friendChatId	friendChatMember
1	["1155169171","1155164941"]
2	["1155164941","1155162650"]
3	["1155169171","1155162650"]

数据下载：https://github.com/Liny777/UserMongoDB_Altas/blob/master/friendChatRoom.json

4、userChatRoom

userId	groupChatRoomId	friendChatRo
"1155169171"	[1]	[1,3]
"1155164941"	[1]	[1,2]
"1155162650"	[1]	[2,3]

数据下载：

https://github.com/Liny777/UserMongoDB_Altas/blob/master/userChatRoom.json

MYSQL

1、user

id	name	passwor
"1155169171"	"Liny"	"123456
"1155164941"	"Alice"	"123456
"1155162650"	"William"	"123456

INSERT INTO user (id,name,password) VALUES ("1155169171","Liny","12345678");

INSERT INTO user (id,name,password) VALUES ("1155164941","Alice","12345678");

INSERT INTO user (id,name,password) VALUES ("1155162650","William","12345678");

select * from user;

```
mysql> select * from user;
+------------+---------+----------+
| id         | name    | password |
+------------+---------+----------+
| 1155162650 | William | 12345678 |
| 1155164941 | Alice   | 12345678 |
| 1155169171 | Liny    | 12345678 |
+------------+---------+----------+
3 rows in set (0.00 sec)
```

2、messages

id	type	chatroom_id	user_id	name（发送者）	mess
1	"friend"	"61c3096a94cc1ad209f6e741"	"1155169171"	"Liny"	"Hel
2	"friend"	"61c36d2194cc1ad209f6e744"	"1155164941"	"Alice"	"Hi"
3	"friend"	"61c36d3c94cc1ad209f6e745"	"1155162650"	"William"	"Tha

INSERT INTO messages (id,type,chatroom_id,user_id,name,message,message_time)

VALUES

(1,"friend","61c3096a94cc1ad209f6e741","1155169171","Liny","Hello",NOW());

INSERT INTO messages (id,type,chatroom_id,user_id,name,message,message_time)

VALUES (2,"friend","61c36d2194cc1ad209f6e744","1155164941","Alice","Hi",NOW());

INSERT INTO messages (id,type,chatroom_id,user_id,name,message,message_time)

VALUES

(3,"friend","61c36d3c94cc1ad209f6e745","1155162650","William","Thanks",NOW());

select * from messages;

```
mysql> select * from messages;
+----+--------+------------+------------+---------+---------+---------------------+
| id | type   | chatroom_id | user_id    | name    | message | message_time        |
+----+--------+------------+------------+---------+---------+---------------------+
|  1 | friend |           1 | 1155169171 | Liny    | Hello   | 2021-12-23 01:55:56 |
|  2 | friend |           2 | 1155164941 | Alice   | Hi      | 2021-12-23 01:57:19 |
|  3 | friend |           3 | 1155162650 | William | Thanks  | 2021-12-23 01:57:26 |
+----+--------+------------+------------+---------+---------+---------------------+
3 rows in set (0.00 sec)
```

3、pushToken

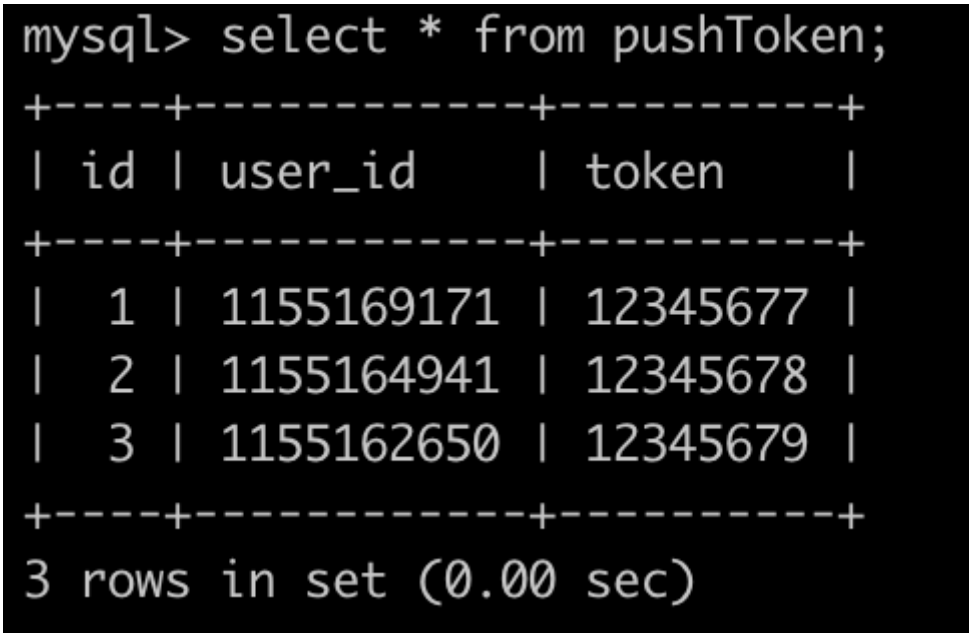
id	**user_id（User 表里的 id）**	**toke**
1	"1155169171"	"123
2	"1155164941"	"123
3	"1155162650"	"123

INSERT INTO pushToken (id,user_id,token) VALUES (1,"1155169171","12345677");

INSERT INTO pushToken(id,user_id,token) VALUES (2,"1155164941","12345678");

INSERT INTO pushToken (id,user_id,token) VALUES (3,"1155162650","12345679");

select * from pushToken;



□ 相关链接

1、MongoDB Altas 教程及优缺点说明：https://zhuanlan.zhihu.com/p/98916948

2、MongoDB 程: https://github.com/Liny777/UserMongoDB_Altas/blob/master/Lab_4.pdf

3、Socket 参考教程：https://github.com/jCzachor6/questions-app-spring-websocked-android-client

4、Websocket vs STOMP : https://stackoverflow.com/questions/40988030/what-is-the-difference-between-websocket-and-stomp-protocols

5、Android socket : https://stackoverflow.com/questions/67665222/android-studio-connect-to-socket-server

6、FCM&SpringBoot ： https://writecodewithprince.com/send-push-notification-with-firebase-in-spring-boot-api/

7、MYSQL: https://segmentfault.com/a/1190000016374807

8、MongoDB responsiblity : https://chikuwa-tech-study.blogspot.com/2021/05/spring-boot-mongorepository.html

常用命令

Mysql

1、查看数据库：show databases;

2、使用数据库：use databses;

3、删除表：Drop table “table_name"

4、登陆数据库：`sudo mysql -u root -p`

5、 DELETE FROM user WHERE id=1;

遇到问题

安卓

Q1、Cannot resolve symbol AppCompatActivity - Support v7 libraries aren't recognized?

A1:Sometimes clearing the android studio caches help.

In android studio I just cleared the caches and restarted with the following option--

File->Invalidate Caches/Restart

Q2、Error: Unfortunately you can't have non-Gradle Java modules and > Android-Gradle modules in one project

A2、First of all you should update to Android Studio 1.2 Source: https://code.google.com/p/android/issues/detail?id=77983

Then you should go to File -> Invalidate Caches / Restart -> Invalidate Caches & Restart.

Then try to build the application again.

or

1- close the project

2- close

Android Studio

IDE

3- delete the

.idea

directory located inside the project folder

4- delete all

.iml

files

5- open

Android Studio

IDE and import the project

服务器

Q1: org.apache.shiro 与 org.springframework.boot 的 spring-boot-starter-web 区别

A：好像是通过这种方式引入了 Shiro 安全框架，Shiro 用 starter 方式优雅整合到 SpringBoot 中，spring-boot-starter-web：web 的场景，自动帮我们引入了 web 模块开发需要的相关 jar 包，起步依赖的核心就是对 pom 文件的配置优化。Spring boot 版本升级的过程也是一个 pom 配置不断优化的过程。——来自 yg 菜鸡的回答，期待贺神的更改。

Q2: @Repository 的作用

Q3: 连不上远程 mysql

1、检查端口是否开放、防火墙是否关闭 ufw status

2、root 身份登录数据库，检查 user 表中对应的用户 host 是否为“%”，修改后 flush previlege

```
mysql> select user,host from user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| alice            | %         |
| debian-sys-maint | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
```

3、注释掉/etc/mysql/mysql.conf.d/mysqld.cnf 中 bind-address=127.0.0.1 一行并重启 mysql service mysql restart