前端react+axios+websocket+antd+scss

后端go+gin+gorm+mysql+redis+websocket+jwt

后端所有的服务失败（mysql，redis，websocket，token）都会返回前端迫使其跳转回登录页

# 前端

登录页-注册页-个人详情页-好友列表页-私聊页-在线交流大厅页

注册前端校验两次密码是否一直，注册成功跳回登录页，登陆成功后存储userID以及后端传来的token到localstarage

对除了登录页和注册页所有的页面以及页面内请求接口都做了jwt校验，如果localstorage中没存token或token后端校验后已过期，则跳转回登录页（这里是为了练习前端，真正使用JWT校验时应该在后端加拦截器。前端每次请求携带token，后端进行校验，过期了则提示前端跳回登录页，否则刷新token）

在个人详情页可以新建或更新个人信息，对性别和电话号码字段进行校验。

好友列表页可以进行新增删除好友，以及进入交流大厅，私聊，个人详情页的功能。

在线交流大厅通过websocket的query只传递userid，此时代表实时群聊，监听连接的onmessage返回，将消息增添至列表，发送消息调用onSend

好友列表页通过websocket的query传递userId，friendId，此时代表私聊，同时调用接口获取好友历史发来的消息，以及历史发向好友的消息，前端会暂存上述返回的历史消息以及实时发送和获取的消息，但在离开页面后清除。

好友列表页可以进入好友的个人主页查看，这里后端采用redis缓存，第一次查询时走数据库，随后将用户详情信息存储在redis，设置过期时间（防止缓存雪崩，这里采用10分钟+1分钟内的随机数过期），在用户更新个人详情时，覆盖redis内容，同时防止缓存穿透，这里采用暂时设置假数据，如果第一次查数据库中没有，则在redis中存一个暂时为空1分钟的假数据。

# 后端

Mysql中建立三个表，users用来存储用户信息（id，userId，psw）等，userDetails用来存储用户登录后设置的个人详情信息，防止泄露密码。通过gorm的多对多关系自动创建好友关系表user-friends

注册后端校验用户名是否已存在，登录页校验用户-密码是否能在数据库中找到匹配，查询成功则通过JWT（密钥为固定的string）创建一个有效期为10min的token返回前端，前端每次请求接口（除了登陆注册）都会回传token用于校验当前登录是否有效。

UserDetail不是必须，查询详情时，通过userId找到则返回，找不到返回找不到，支持创建/修改。

通过前端连接websocket时是否传递了friendId（群聊不需要指定friendId）判断是否是群聊，群聊通过websocket保存当前所有接入客户端的conn，任一方发送消息，则遍历connMap进行广播（剔除自身），connMap存储conn及userId的键值对。消息不做持久化存储，离开会议室消息消失。

前端连接websocket时传递了friendId则是私聊，私聊开始前通过接口将redis存储的历史消息（user-friend及friend-user）返回前端。Redis在启动后端服务前要提前启动（go代码只是和redis建立连接），每当websocket服务中台收到私聊消息时，先通过userId-friendId的方式存在redis的list中，每个list最多只保存5条消息，超出则丢弃最早的消息；再遍历connMap寻找匹配friendId的conn进行传递消息，未找到代表对方未上线。

获取好友列表直接查表

添加好友后端查表，查到后通过gorm多对多创建好友关系，双向创建（a是b的好友，同时b也是a的好友），查找失败向前端返回失败。

删除好友通过gorm多对多删除好友关系，双向删除