

前端 技术路线 架构

Flutter 完成界面开发

models	2	✓
pages	4	✓
screens	2	✓
widgets	4	✓
main.dart	537 B	✓
theme.dart	3.27 K	✓

- models 需要使用的抽象模型
- pages 页面
- screens 屏幕
- widgets 组件复用
- theme.dart 亮色深色需要使用的主题色

连接部分

- dio 模块 异步请求接口
- socket.io 建立 websocket 持久链接

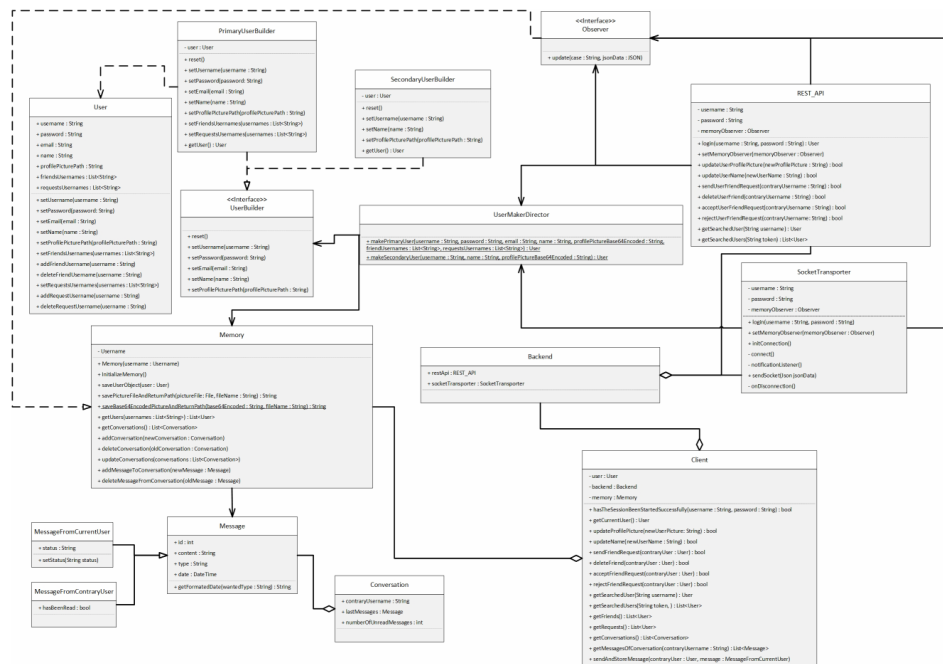
后端 技术路线 架构

- 使用 `openapi-generator` 生成 `django` 接口，手动调试
- `django-restfulframework`, `websocket`

API 设计

User		▼
GET	/login login	
GET	/userinfo userinfo	
GET	/getallusers getallusers	
GET	/user/updateUserName updateUserName	
GET	/user/updateUserProfilePicture updateUserProfilePicture	
GET	/updateUserPwd updateUserPwd	
GET	/searchuserbyname searchuserbyname	
GET	/register register	
friend		▼
GET	/deletefriend deletefriend	
GET	/myfriends myfriends	
Group		▼
GET	/groupinfo groupinfo	
GET	/mygroup mygroup	
GET	/joingroup joingroup	
GET	/creategroup creategroup	
GET	/searchgroupbyname searchgroupbyname	

数据库设计



完成进度

- 前端完成主屏幕 四个基础页面, message 页面基本完成
- 完成几个组件的分离, 作为以后复用
- 导入 `faker` 包 作为 mock 数据使用
- 到与接口对接时, 组件可能需要重写完成异步请求
- 后端进度0, 找不到可参考资料, 大多为 调用 sdk, 使用 firebase 等

