README.md

Technology:

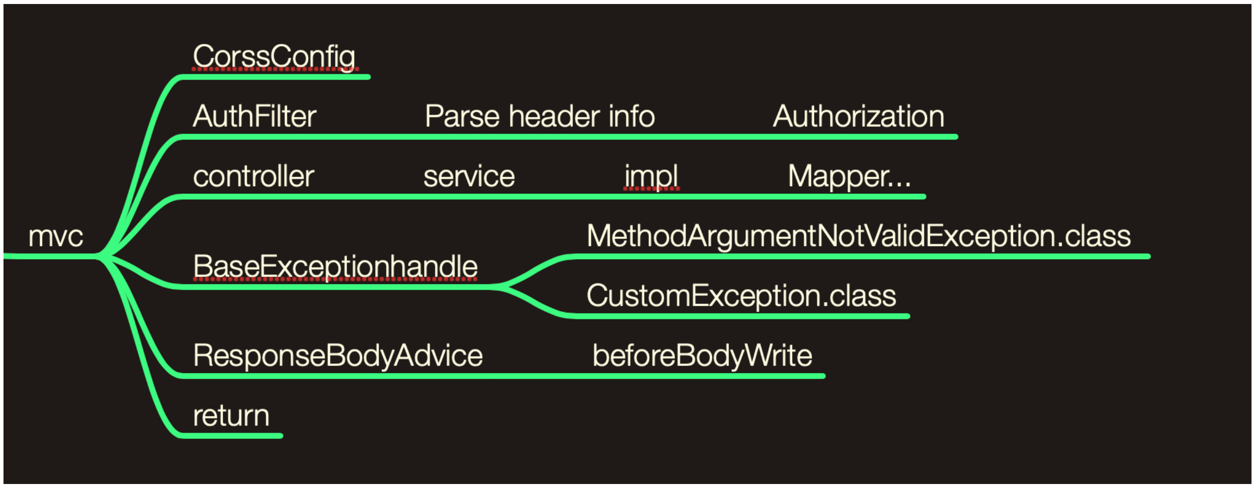
sprintboot + mybatis + redis

API doc:

https://github.com/NominationP/wiredcraft/blob/main/README.md

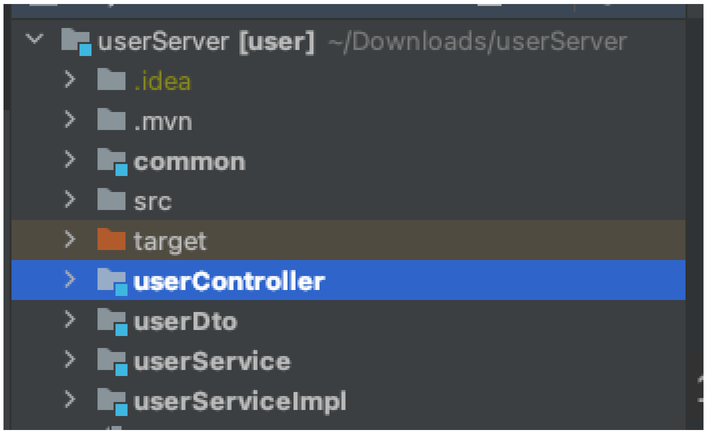
Process：

MVC：



Impress

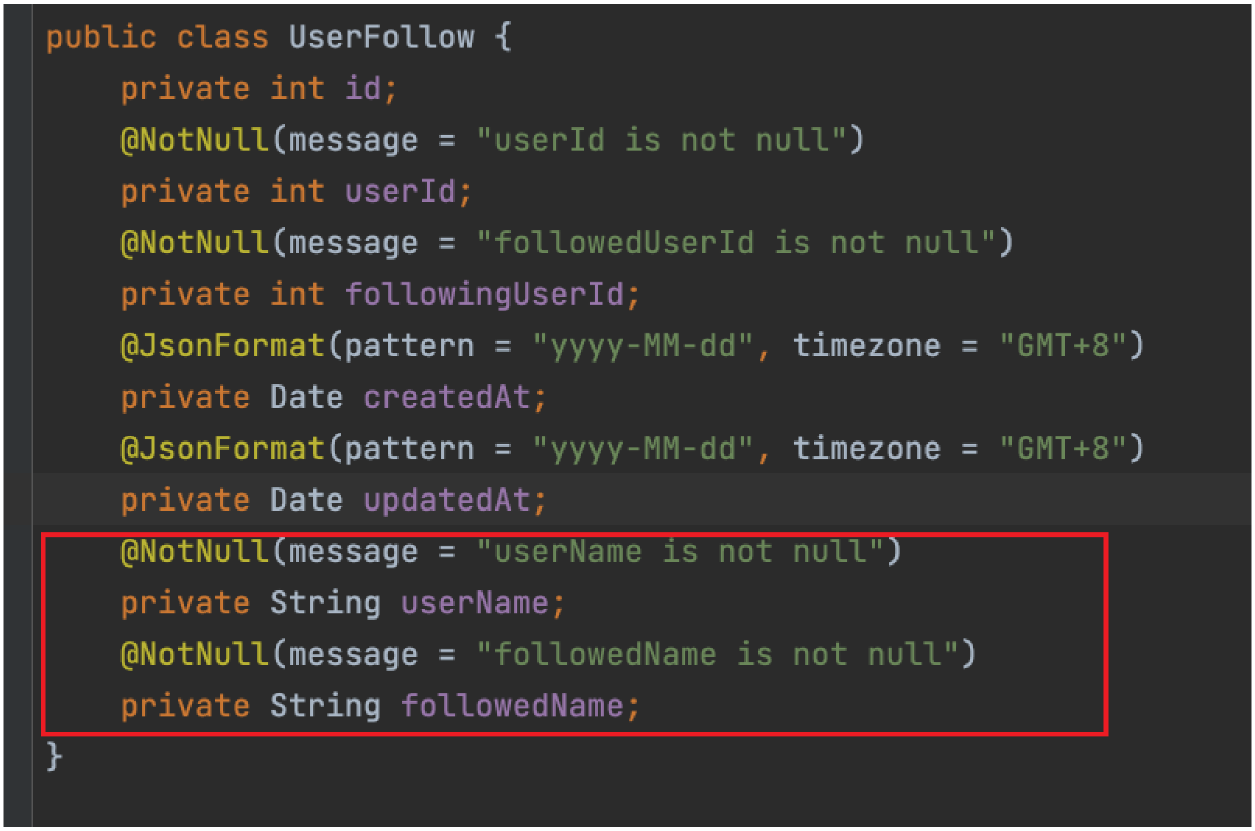
structure



```
userServer [user] ~/Downloads/userServer
  .idea
  .mvn
  common
  src
  target
  userController
  userDto
  userService
  userServiceImpl
```

- Q1 Why use multi modules ?

Mysql redundancy design



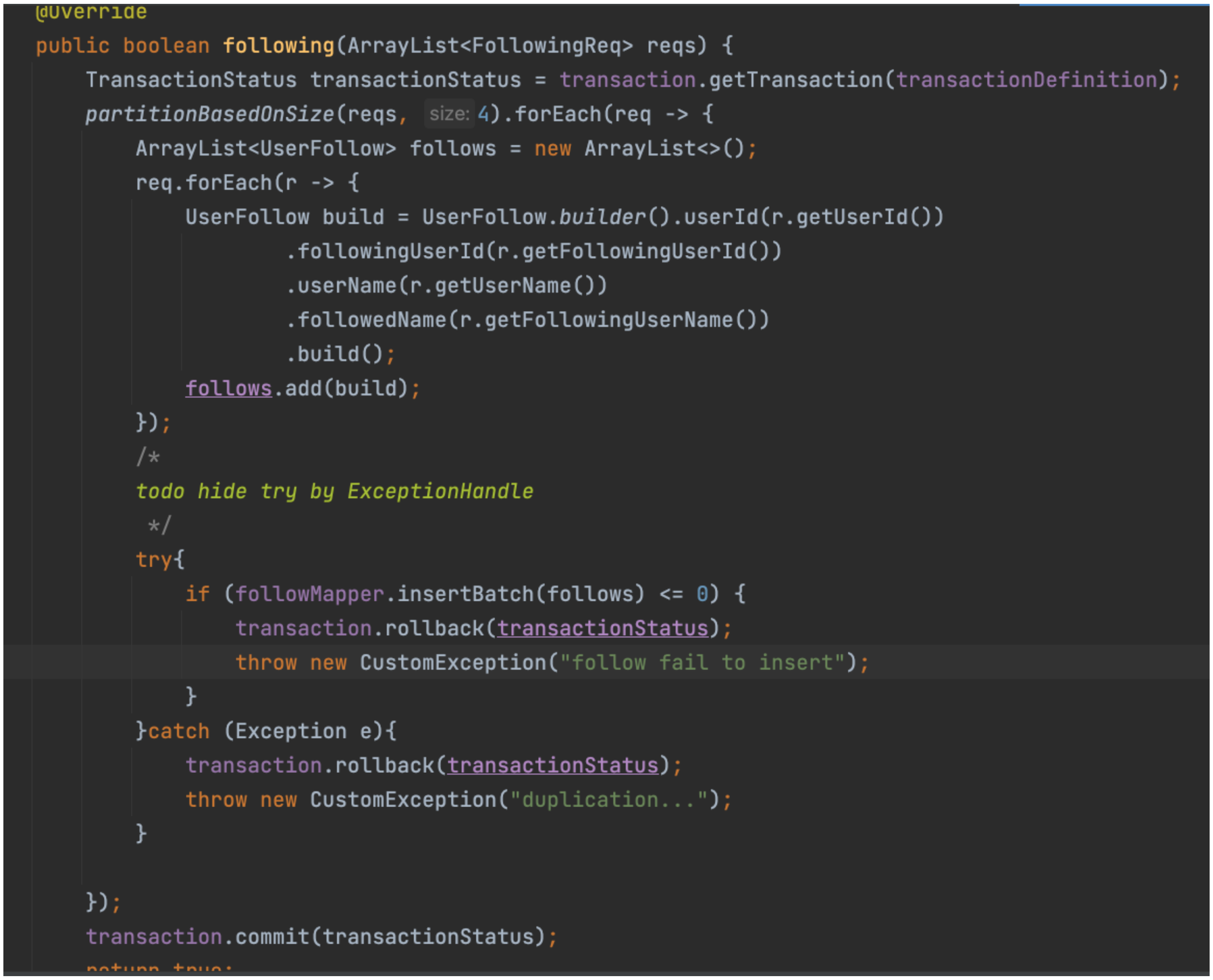
```java
public class UserFollow {
    private int id;
    @NotNull(message = "userId is not null")
    private int userId;
    @NotNull(message = "followedUserId is not null")
    private int followingUserId;
    @JsonFormat(pattern = "yyyy-MM-dd", timezone = "GMT+8")
    private Date createdAt;
    @JsonFormat(pattern = "yyyy-MM-dd", timezone = "GMT+8")
    private Date updatedAt;
    @NotNull(message = "userName is not null")
    private String userName;
    @NotNull(message = "followedName is not null")
    private String followedName;
}
```

Aouth2

UnitTest

- Q1 Why so few unit tests

Transaction



```java
@Override
public boolean following(ArrayList<FollowingReq> reqs) {
    TransactionStatus transactionStatus = transaction.getTransaction(transactionDefinition);
    partitionBasedOnSize(reqs, size: 4).forEach(req -> {
        ArrayList<UserFollow> follows = new ArrayList<>();
        req.forEach(r -> {
            UserFollow build = UserFollow.builder().userId(r.getUserId())
                    .followingUserId(r.getFollowingUserId())
                    .userName(r.getUserName())
                    .followedName(r.getFollowingUserName())
                    .build();
            follows.add(build);
        });
        /*
        todo hide try by ExceptionHandle
         */
        try{
            if (followMapper.insertBatch(follows) <= 0) {
                transaction.rollback(transactionStatus);
                throw new CustomException("follow fail to insert");
            }
        }catch (Exception e){
            transaction.rollback(transactionStatus);
            throw new CustomException("duplication...");
        }

    });
    transaction.commit(transactionStatus);
    return true;
```

- Q1 Why use manual transaction
- Q2 reduce try-catch
- Q3 Notice rollback type