1. 实现功能
2. 用户注册登录
3. 用户策略信息
4. 用户交易账户登录
5. 用户交易信息
6. 行情订阅信息
7. 下单/撤单操作
8. 系统日志

* Mysql数据库存储数据如下
* 用户基本信息（用户名、密码、手机号等）
* 操作日志
* 其他信息
* Redis存储：
* 策略信息（策略名称、策略标的、开始时间、结束时间）
* 行情订阅信息

1. 功能详解
2. 用户注册登录
3. 用户信息包含：用户ID、用户名、密码、手机号、邮箱、登录IP、登录时间、注册时间等
4. 注册时发送手机验证码
5. 用户信息保存至Mysql
6. 用户策略信息
7. 策略信息包含：策略Id、策略名、策略创建人Id、策略创建时间、结束时间、策略状态、策略合约（组合）、策略标的组合中每个合约的下单价格、下单数量、平仓价格、平仓数量
8. 策略基本数据保存至Mysql
9. 策略运行数据保存至Redis
10. 用户交易账户登录
11. 用户可以有多个交易账户
12. 信息保存于MySql中
13. 包含信息：接口Id、接口名称、经纪商Id、用户ID、密码（不保存）等，登录交易账户时，提示用户输入密码
14. 用户交易信息

下单服务器返回交易数据，直接保存于Redis，然后通过中控服务器传递至网页上显示

1. 行情订阅信息

用户订阅行情后，由行情服务器进行推送

1. 下单/撤单操作

针对手工操作，发送相对应的指令至中控服务器

1. 系统日志

每一个操作都要记录日志，保存至Mysql

1. 具体实现
2. 数据库表
3. 用户表
4. 功能模块表
5. 日志表
6. 交易所表
7. 商品表
8. 合约表
9. 策略表
10. 策略----功能表
11. 交易账户表

2、实现方式

（1）功能列表使用树菜单实现

（2）web服务器通过websocket/socket方式将行情及交易数据传递给页面

（3）web服务器与中控服务器通过Netty传递数据

* 中控服务器推送相应Web服务器的数据
* Web服务器自行解析并分发至网页

注：需要调研如何通过websocket/socket给特定的页面（用户）发送数据

1. web服务器推送给网页的数据包含行情、日志、成交数据（委托、持仓、成交等）

* 行情数据比较密集，可开启单独的websocket连接
* 其他数据可使用同一websocket连接来进行区分