

体系结构设计文档

体系结构设计文档

- 更新历史
- 1、引言
- 2、逻辑视角
 - 2.1、分层架构图
 - 2.2、逻辑包图
- 3、组合视图
 - 3.1、物理包的划分
 - 3.2、物理包图
- 4、接口视图
 - 4.1、模块的职责
 - 4.2、用户界面模块的接口规范
 - 4.2.1、用户界面模块的分解
 - 4.2.2、业务逻辑模块的分解
 - 4.2.2.1、业务逻辑层模块的职责
 - 4.2.2.2、业务逻辑层模块接口规范
 - 4.2.2.2.1供接口
 - 4.2.2.2.2需接口
 - 4.2.3、数据层的分解
 - 4.2.3.1、数据层模块的职责
 - 4.2.3.2、数据层模块的接口规范
 - 4.2.3.2.1供接口
- 5、信息视角
 - VO定义
 - PO定义
 - Exception定义

更新历史

修改人员	日期	变更原因	版本号
黄泽鹏	2017/5/28	构建基本框架	V1.0
刘宇翔	2017/5/30	数据层供接口	V1.1
黄泽鹏	2017/6/2	VO、PO、Exception定义	V1.2

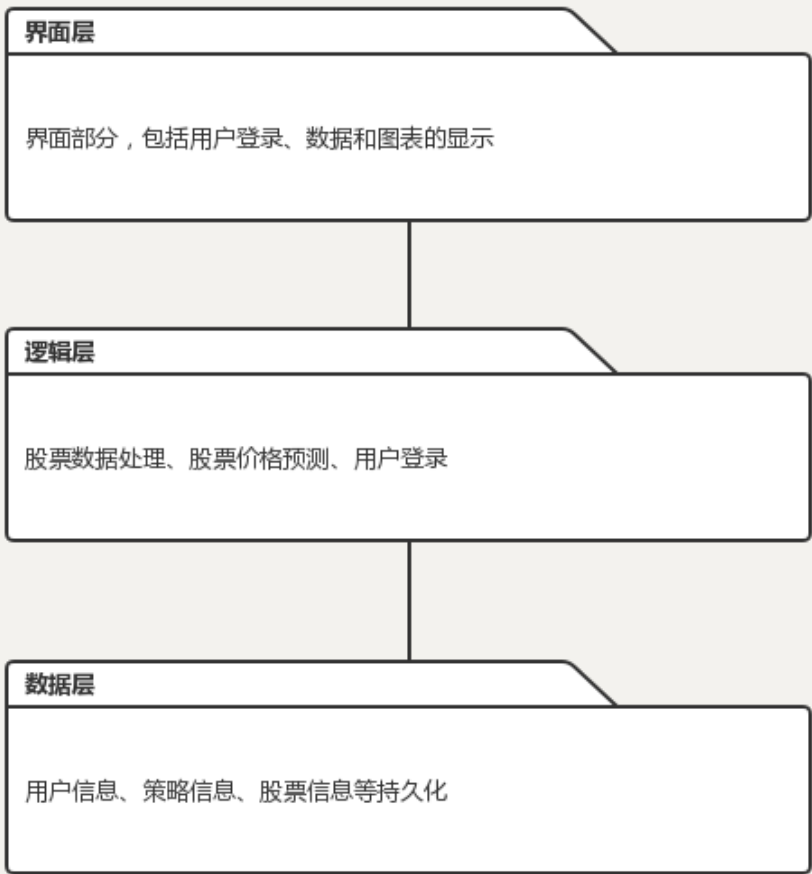
1、引言

编

写 目 的	本文档提供股票信息分析系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。
对 象 和 范 围	本文档的读者是 Eureka 团队内部的开发和管理人员,参考了《软件工程与计算（三）》中的《Sharing Team软件架构设计文档片段》，用于指导下一循环的代码开发和测试工作。
参 考 文 献	《项目启动文档》《软件需求规格说明书》《软件架构文档模板》《软件工程与计算（卷三）》
名 词 和 术 语	

2、逻辑视角

2.1、分层架构图



2.2、逻辑包图

3、组合视图

3.1、物理包的划分

开发（物理包）	依赖的其他开发包
dataController	
browseStrategyUI	dataController, logic, chart, mainScreen marketUI, rmi, vo
chart	vo, en_um
loginUI	dataController, exception, logic, rmi
mainScreen	dataController, browseStrategyUI, marketUI stockSetUI, strategyUI, rmi, vo

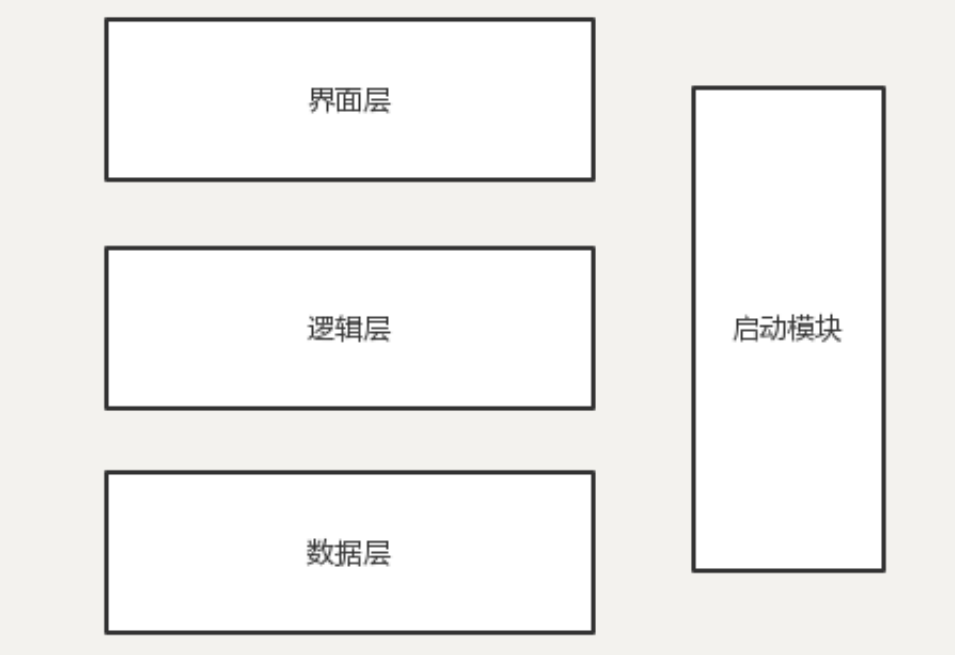
marketUI	dataController, exception, logic, rmi vo, chart, mainScreen, singleStockUI
saveAsPNG	
singleStockUI	dataController, exception, logic, en_um chart, saveAsPNG, rmi, vo
statisticsUI	dataController, exception, logic, en_um chart, rmi, vo
stockSetUI	dataController, exception, logic, en_um chart, rmi, vo
strategyUI	dataController, exception, logic, chart marketUI, saveAsPNG, rmi, vo
rmi	exception, logic, vo, po, data
logic	data, exception, vo, po
data	exception, po
vo	po
po	

3.2、物理包图

4、接口视图

4.1、模块的职责

客户端模块如图所示。客户端各层职责由表2所示。



层	职责
启动模块	负责初始化网络通信机制，启动用户界面
界面层	基于窗口的股票信息分析系统客户端用户界面
逻辑层	对于用户界面的输入进行响应并进行业务处理
数据层	接受逻辑层请求并调用数据库获取数据

接口	服务调用方	服务提供方
ClientLogicInterface ForecastRODInterface StockLogicInterface	界面层	逻辑层
IDataInterface	逻辑层	数据层

4.2、用户界面模块的接口规范

4.2.1、用户界面模块的分解

模块	职责
需要的服务（需接口）	

服务名	服务
ClientLogicInterface	用户登录和退出
ForecastRODInterface	对股票明天价格的预测
StockLogicInterface	对于股票和市场信息的获取

4.2.2、业务逻辑模块的分解

4.2.2.1、业务逻辑层模块的职责

模块	职责
ClientLogicInterface	用户登录模块
ForecastRODInterface	对股票明天价格的预测
StockLogicInterface	对于股票和市场信息的获取

4.2.2.2、业务逻辑层模块接口规范

4.2.2.2.1供接口

提供的服务（供接口）		
ClientLogicInterface.signUp	语 法	public void signUp (String username, char[] password) throws RemoteException, UserNameRepeatException;
	前置条件	无
	后置条件	注册新用户
ClientLogicInterface.signIn	语 法	public void signIn (String username, char[] password) throws RemoteException, LogErrorException, SqlNotConnectedException;
	前置条件	用户已注册
已 经 用 户 登 录		

后
置
条
件

登录用户账户

ClientLogicInterface.signOut

语
法

public void signOut(String username)
throws RemoteException;

前
置
条
件

用户已登录

后
置
条
件

登出用户

提供的服务（供接口）

StockLogicInterface.
getExponentChart

语
法

public ExponentChartVO
getExponentChart(String name,LocalDate
begin,LocalDate end)

前
置
条
件

无

后
置
条
件

无

StockLogicInterface.
getExponentEMAInfo

语
法

public List< EMAInfoVO> getExponentEMA
String name, LocalDate begin, LocalDate e

前
置
条
件

无

后
置
条
件

无

StockLogicInterface.
getLatestCommpanyInfo

语
法

public CompanyInfoVO
getLatestCommpanyInfo(LocalDate time,
code) throws NullStockIDException,
NullDateException, RemoteException;

前
置
条
件

	置 条 件	
	后 置 条 件	
StockLogicInterface. getExponentInfoByTime	语 法	List< SingleStockInfoVO> getExponentInf (String name, LocalDate begin, LocalDate
	前 置 条 件	无
	后 置 条 件	无
StockLogicInterface. getSingleStockInfoByTime	语 法	List< SingleStockInfoVO> getSingleStockInfoByTime (String stockC LocalDate begin, LocalDate end)
	前 置 条 件	无
	后 置 条 件	无
StockLogicInterface. getEMAInfo	语 法	List< EMAInfoVO> getEMAInfo (String st LocalDate begin, LocalDate end)
	前 置 条 件	无
	后 置 条 件	无
StockLogicInterface. getComparedInfo	语 法	ComparedInfoVO getComparedInfo (Stri stockCodeA, LocalDate begin, LocalDate e
	前 置 条 件	无

	置 条 件	
	后 无 置 条 件	
StockLogicInterface. getMarketInfo	语 法 MarketInfoVO getMarketInfo (LocalDate date,String marketname)	
	前 工作日 置 条 件	
	后 无 置 条 件	
StockLogicInterface. getExponentEMAInfo	语 法 public EMAInfoVO getExponentEMAInfo(name, LocalDate begin, LocalDate end ,int	
	前 无 置 条 件	
	后 无 置 条 件	
StockLogicInterface. getIndustryList	语 法 List< String> getIndustryList() throws RemoteException;	
	前 无 置 条 件	
	后 无 置 条 件	
StockLogicInterface. getStockSet	语 法 List< String> getStockSet(String username RemoteException;	
	前 无 置	

	条件
	后置条件
StockLogicInterface. getStockSetSortedInfo	语法 List< SingleStockInfoVO> getStockSetSortedInfo(String stockSetName,LocalDate now,String username) throws RemoteException, NullMarketException;
	前置条件
	后置条件
StockLogicInterface. getStockBasicInfo	语法 SingleStockInfoVO getStockBasicInfo(String stockID,LocalDate now) throws NullStockIDException, NullDateException, RemoteException;
	前置条件
	后置条件
StockLogicInterface. setStrategy	语法 void setStrategy(StrategyConditionVO strategyConditionVO, SaleVO s, LocalDate now, String stockSetName,String username)
	前置条件
	后置条件 生成策略
StockLogicInterface. getYieldChartData	语法 YieldChartDataVO getYieldChartData(String stockID,LocalDate start,LocalDate end)

StockLogicInterface. addStockToStockSet	语 法	void addStockToStockSet(String stockName, String stockSetName, String username) throws StockNameRepeatException, RemoteException;
	前 置 条 件	有股票池
	后 置 条 件	无
StockLogicInterface. deleteStockFromStockSet	语 法	void deleteStockFromStockSet(String stockName, String stockSetName, String username) throws RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. nameToCode	语 法	String nameToCode(String name) throws RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. saveStrategy	语 法	void saveStrategy(StrategyVO strategyVO, String username) throws RemoteException, StrategyRepeatException;
	前 置 条 件	
	后 置 条 件	

	件	
StockLogicInterface. deleteStrategy	语 法	void deleteStrategy (String createName, S strategyName) throws RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. comment	语 法	void comment(String Username, String strategyName, String commenterName, LocalDateTime time, String comment) thr RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. getStrategy	语 法	StrategyShowVO getStrategy (String crea String StrategyName) throws RemoteExc
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. getStrategyList	语 法	List< StrategyListVO> getStrategyList (St creatorName) throws RemoteException;
	前 置 条 件	
	后 置 条 件	

	件	
StockLogicInterface. getStrategyList	语 法	List< StrategyListVO> getStrategyList ()th RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. setPublic	语 法	void setPublic(String creatroName, String straetgyName,boolean property) throws RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. getAreaList	语 法	List< String> getAreaList() throws RemoteException;
	前 置 条 件	
	后 置 条 件	
StockLogicInterface. getConceptList	语 法	List< String> getConceptList() throws RemoteException;
	前 置 条 件	
	后 置 条 件	

StockLogicInterface. getTimeSharingData	语 法	TimeSharingVO getTimeSharingData(String code,LocalDate date)throws TimeShraingLackException,NullStockIDException;
	前置 条件	
	后置 条件	
StockLogicInterface. fuzzySearch	语 法	List< String> fuzzySearch(String input) throws RemoteException;
	前置 条件	
	后置 条件	

4.2.2.2.2需接口

需要的服务（需接口）		
StockDataService.getTimeSharingData	语 法	public List getTimeSharingData(String code,LocalDate date)throws TimeShraingLackException,NullStockI
	前置 条件	存在的交易日和股票编号
	后置 条件	获得股票的分时图
StockDataService.addBrowseTimes	语 法	public void addBrowseTimes(String stc throws NullStockIDException
	前	股票编号
	后	

	置 条 件
	后 无 置 条 件
StockDataService.fuzzySearch	语 public List fuzzySearch(String code) 法
	前 用户输入 置 条 件
	后 存在该字段的搜索结果 置 条 件
StockDataService.getMinDay	语 public LocalDate getMinDay(String sto 法 throws NullStockIDException
	前 存在的股票编号 置 条 件
	后 该股票最早一日交易日 置 条 件
StockDataService.getMaxDay	语 public LocalDate getMaxDay(String stc 法 throws NullStockIDException
	前 存在的股票编号 置 条 件
	后 该股票最晚一日交易日 置 条 件
StockDataService.getSingleStockInfo	语 public SingleStockInfoPO 法 getSingleStockInfo(String stockcode, L date) throws NullStockIDException, NullDateException
	前 存在的股票编号和交易日 置 条 件

后 该日该股票情况

置条件

StockDataService.addDays

语 public LocalDate addDays(LocalDate d
法 last,boolean zheng) throws DateOverE

前 正确的时间

置条件

后 从该日起向前后推x天的日期

置条件

需要的服务（需接口）

StockSet.getStockSet

```
public List getStockSet(String
username)
```

前 存在的用户名

置条件

后 该用户所有股票池

置条件

StockSet.addStockSet

```

语 public void addStockSet(String
法 stockSetName, String username)
   throws
   StockSetNameRepeatException

```

前 存在的用户名，不重复的股票池名
置 字

条件

后 添加一个新的股票池

置条件

StockSet.deleteStockSet	语 法 public void deleteStockSet(String stockSetName, String username) 前 置 条 件 存在的用户名，股票池名字 后 置 条 件 删除一个股票池
StockSet.addStockToStockSet	语 法 public void addStockToStockSet(String stockName, String stockSetName, String username) throws StockNameRepeatException 前 置 条 件 存在的用户名，股票池名字，不重复的股票编号 后 置 条 件 添加一个新的股票到股票池
StockSet.deleteStockFromStockSet	语 法 public void deleteStockFromStockSet(String stockName, String stockSetName, String username) 前 置 条 件 存在的用户名，股票池名字，股票编号 后 置 条 件 删除一个股票从股票池
StockSet.getStockSetInfo	语 法 public List getStockSetInfo(String stockSetName,String userName) 前 置 条 件 存在的用户名，股票池名字 后 置 条 件 股票池中所有股票编号

	置 条 件
StockSet.getStockSetInfo	语 法 public List getStockSetInfo(String stockSetName)
	前 置 条 件 存在的股票池名字
	后 置 条 件 股票池中所有股票编号
StockSet.getIndustryList	语 法 public List getIndustryList()
	前 置 条 件 无
	后 置 条 件 所有行业板块名字
StockSet.getConceptList	语 法 public List getConceptList()
	前 置 条 件 无
	后 置 条 件 所有概念板块名字
StockSet.getAreaList	语 法 public List getAreaList()
	前 置 条 件 无
	后 置 条 件 所有地域板块名字

需要的服务（需接口）

strategy.saveStrategy 语 public void
法 saveStrategy(StratgyInfoPO po,String
strategyName,String username) throws
StrategyRepeatException

前 存在的用户名，配置信息，策略名字
置
条
件

后 存储一个策略
置
条
件

strategy.deleteStrategy 语 public void deleteStrategy (String
法 createName, String strategyName)

前 存在的用户名，策略名字
置
条
件

后 删除一个策略
置
条
件

strategy.comment 语 public void comment(String Username,
法 String strategyName, String
commenterName, LocalDateTime time,
String comment)

前 创建者名字，策略名字，评论者名字，
置 评论，时间
条
件

后 评论一个策略
置
条
件

strategy.getStrategy 语 public StrategyShowPO getStrategy (
法 String creatorName, String
StrategyName)

前 创建者名字，策略名字
置

	条件
	后置条件 获得一个策略的详细信息
strategy.applyStrategy	语 public StrategyInfoPO 法 applyStrategy(String creatorName,String strategyName)
	前置条件 创建者名字，策略名字
	后置条件 应用一个策略的方案
strategy.addStrategyShow	语 public void addStrategyShow (String 法 creatorName, String StrategyName ,StrategyShowPO vo)
	前置条件 创建者名字，策略名字，策略详细信息
	后置条件 存储一个策略的详细信息
strategy.getStrategyComments	语 public List 法 getStrategyComments(String creatorName,String strategyName)
	前置条件 创建者名字，策略名字
	后置条件 这个策略的所有评价
strategy.clearStrategyShow	语 public void clearStrategyShow () 法
	前置条件 无
	后置条件

	置 条 件
	后 清空所有策略详细信息 置 条 件
strategy.getStrategyList	语 public List getStrategyList (String 法 creatorName)
	前 创建者名字 置 条 件
	后 获得该用户所有策略 置 条 件
strategy.getStrategyList	语 public List getStrategyList () 法
	前 无 置 条 件
	后 获得所有公开策略 置 条 件
strategy.setPublic	语 public void setPublic(String 法 creatroName, String straetgyName,boolean property)
	前 创建者名字，策略名字，是否公开 置 条 件
	后 修改一个策略的公开属性 置 条 件
需要的服务（需 接口）	
user signInCheck	语 public void signInCheck(String username, String

user.signUpCheck	方法	public void signUpCheck(String username, String password) throws UserNameRepeatException
	前置条件	不重复的用户名与密码
	后置条件	注册一个新账号
user.signInCheck	方法	public void signInCheck(String username, String password) throws LogErrorException
	前置条件	用户名与密码
	后置条件	登录该用户
user.logout	方法	public void logout(String username)
	前置条件	用户名
	后置条件	登出该用户

4.2.3、数据层的分解

4.2.3.1、数据层模块的职责

模块	职责
----	----

4.2.3.2、数据层模块的接口规范

4.2.3.2.1供接口

提供的服务（供接口）		
StockDataService.getTimeSharingData	语 法	public List getTimeSharingData(String code,LocalDate date)throws TimeShraingLackException,NullStockI
	前 置 条 件	存在的交易日和股票编号
	后 置 条 件	获得股票的分时图
StockDataService.addBrowseTimes	语 法	public void addBrowseTimes(String sto throws NullStockIDException
	前 置 条 件	股票编号
	后 置 条 件	无
StockDataService.fuzzySearch	语 法	public List fuzzySearch(String code)
	前 置 条 件	用户输入
	后 置 条 件	存在该字段的搜索结果
StockDataService.getMinDay	语 法	public LocalDate getMinDay(String sto throws NullStockIDException
	前 置 条 件	存在的股票编号
	后 置 条 件	该股票最早一日交易日

StockDataService.getMaxDay	语 法	public LocalDate getMaxDay(String stockcode, LocalDate date) throws NullStockIDException
	前置 条件	存在的股票编号
	后置 条件	该股票最晚一日交易日
StockDataService.getSingleStockInfo	语 法	public SingleStockInfoPO getSingleStockInfo(String stockcode, LocalDate date) throws NullStockIDException, NullDateException
	前置 条件	存在的股票编号和交易日
	后置 条件	该日该股票情况
StockDataService.addDays	语 法	public LocalDate addDays(LocalDate date, int days, boolean zheng) throws DateOverException
	前置 条件	正确的时间
	后置 条件	从该日起向前后推x天的日期
提供的服务（供接口）		
StockSet.getStockSet	语 法	public List getStockSet(String username)
	前置 条件	存在的用户名
	后置 条件	该用户所有股票池

	置 条 件	
StockSet.addStockSet	语 法	public void addStockSet(String stockSetName, String username) throws StockSetNameRepeatException
	前 置 条 件	存在的用户名，不重复的股票池名字
	后 置 条 件	添加一个新的股票池
StockSet.deleteStockSet	语 法	public void deleteStockSet(String stockSetName, String username)
	前 置 条 件	存在的用户名，股票池名字
	后 置 条 件	删除一个股票池
StockSet.addStockToStockSet	语 法	public void addStockToStockSet(String stockName, String stockSetName, String username) throws StockNameRepeatException
	前 置 条 件	存在的用户名，股票池名字，不重复的股票编号
	后 置 条 件	添加一个新的股票到股票池
StockSet.deleteStockFromStockSet	语 法	public void deleteStockFromStockSet(String stockName, String stockSetName, String username)
	前 置 条 件	存在的用户名，股票池名字，股票编号
	后 置 条 件	删除一个股票从股票池

	前 置 条 件	存在的用户名，股票池名字，股票编号
	后 置 条 件	删除一个股票从股票池
StockSet.getStockSetInfo	语 法	public List getStockSetInfo(String stockSetName,String userName)
	前 置 条 件	存在的用户名，股票池名字
	后 置 条 件	股票池中所有股票编号
StockSet.getStockSetInfo	语 法	public List getStockSetInfo(String stockSetName)
	前 置 条 件	存在的股票池名字
	后 置 条 件	股票池中所有股票编号
StockSet.getIndustryList	语 法	public List getIndustryList()
	前 置 条 件	无
	后 置 条 件	所有行业板块名字
StockSet.getConceptList	语 法	public List getConceptList()
	前 置 条 件	无

	条件
	后置条件 所有概念板块名字
StockSet.getAreaList	语言 public List getAreaList() 前置条件 无
	后置条件 所有地域板块名字
提供的服务（供接口）	
strategy.saveStrategy	语言 public void saveStrategy(Strate gyInfoPO po,String strategyName,String username) throws StrategyRepeatException 前置条件 存在的用户名，配置信息，策略名字
	后置条件 存储一个策略
strategy.deleteStrategy	语言 public void deleteStrategy (String createName, String strategyName) 前置条件 存在的用户名，策略名字
	后置条件 删除一个策略
strategy.comment	语言 public void comment(String Username, String strategyName,String comment)

	法	String strategyName, String commenterName, LocalDateTime time, String comment)
	前置条件	创建者名字，策略名字，评论者名字，评论，时间
	后置条件	评论一个策略
strategy.getStrategy	语法	public StrategyShowPO getStrategy (String creatorName, String StrategyName)
	前置条件	创建者名字，策略名字
	后置条件	获得一个策略的详细信息
strategy.applyStrategy	语法	public StrategyInfoPO applyStrategy(String creatorName,String strategyName)
	前置条件	创建者名字，策略名字
	后置条件	应用一个策略的方案
strategy.addStrategyShow	语法	public void addStrategyShow (String creatorName, String StrategyName ,StrategyShowPO vo)
	前置条件	创建者名字，策略名字，策略详细信息
	后置条件	存储一个策略的详细信息

	条件
strategy.getStrategyComments	语 public List 法 getStrategyComments(String creatorName,String strategyName)
	前置 条件 创建者名字，策略名字
	后置 条件 这个策略的所有评价
strategy.clearStrategyShow	语 public void clearStrategyShow () 法
	前置 条件 无
	后置 条件 清空所有策略详细信息
strategy.getStrategyList	语 public List getStrategyList (String 法 creatorName)
	前置 条件 创建者名字
	后置 条件 获得该用户所有策略
strategy.getStrategyList	语 public List getStrategyList () 法
	前置 条件 无
	后置 条件 获得所有公开策略

	条件
strategy.setPublic	语法 public void setPublic(String creatroName, String straetgyName,boolean property)
	前置条件 创建者名字，策略名字，是否公开
	后置条件 修改一个策略的公开属性

提供的服务（供接口）

user.signUpCheck	语法 public void signUpCheck(String username, String password) throws UserNameRepeatException
	前置条件 不重复的用户名与密码
	后置条件 注册一个新账号
user.signInCheck	语法 public void signInCheck(String username,String password) throws LogErrorException
	前置条件 用户名与密码
	后置条件 登录该用户
user.logout	语法 public void logout(String username)
	前置条件 用户名
	后置条件 退出该用户

后 登出该用户
置
条
件

5、信息视角

VO定义

- **CommentVO:**
comment: 评论内容
commentTime: 评论时间
commenterName: 评论者姓名
- **ComparedInfoVO:**
List< Double>logYieldA: 每个日期的对数收益率
List< LocalDate> date: 对应日期
List< Integer> diantu: 每个收益率的个数
- **EMAInfoVO:**
List< LocalDate> date: 日期
List< Double> EMA: 每个日期对应均值
tianshu: 均值按几日计算
- **MarketInfoVO:**
volume: 每日交易量
zhangting: 涨停股票数
dieting: 跌停股票数
tingpai: 停牌股票数
junzhi: 市场股票价格均值
fangcha: 市场股票价格方差
List< Integer> shanxingtu: 各涨跌幅度股票数 (range:5%)
List< Integer> diantu:各涨跌幅股票数 (range:1%)
- **PredictVO:**
predictPrice: 预测股票价格
predictROD: 预测股票涨跌幅
- **SaleVO:**
tiaocangqi: 调仓期 (default:10)
tiaocangjiage: 调仓价格 (default:收盘价)
num: 持有股票数
- **SingleStockInfoVO:**
name: 股票名称
date: 日期
code: 股票代码
open: 开盘价
close: 收盘价
high: 当日最高价

low: 当日最低价
volume: 当日交易量
adjclose: 复权收盘价
market: 股票所属市场
fudu: 股票涨跌幅
ave: 股票价格均值

- **StockRODVO:**

int[][] wROD: 每周的涨跌幅分布 (range:1%)
int[][] RODw: 涨跌幅的星期分布
int[][] nodata: 每周是否有数据统计
int[][] YesterdayAvgROD: 前一天收盘价相对于均值涨跌幅
int[] zhixin: 收否在置信区间统计
int[] Pos: 预测涨次数+正确次数
int[] Neg: 预测跌次数+正确次数

- **StrategyConditionVO:**

name:
List< Integer> parameter:

- **StrategyListVO:**

creatorName: 策略创建者名字
strategyName: 策略名称
strategyYearReturn: 策略年化收益率

- **StrategyShowVO:**

List< LocalDate> TimeList: 日期
List< Double> BasicReturn: 基准收益率
List< Double> StrategyReturn: 策略收益率
Alpha: α
Beta: β
Sharp: 夏普比率
zuidahuiche: 最大回撤
StrategyYearReturn: 策略年化收益率
jizhunYearReturn: 基准年化收益率
List< CommentVO> comments: 策略评论
Strategyname: 策略名称
publicorPrivate: 公有 (私有)
strategyConditionVO:
saleVO:

- **StrategyVO:**

name: 策略名称
strategyConditionVO:
saleVO:
publicorprivate: 公有 (私有)

- **YieldChartDataVO:**

List< LocalDate> datelist:
List< Double> jizhunlist:
List< Double> celuelist:
alpha: α

beta: β
sharp: 夏普比率
jizhunyearreturn: 基准年化收益率
yearreturn: 年化收益率
zuidahuiche: 最大回撤

- **YieldDistributionHistogramDataVO:**
Map> zuhe:

PO定义

- **CommentPO:**
creatorName: 创建者名字
strategyName: 策略名称
time: 评论时间
commentUserName: 评论者名字
comments: 评论内容
- **CompanyInfoPO:**
code: 股票代码
date: 截止日期
basicIncome: 基本每股收益
netAsset: 每股净资产
totalCapitalization: 总股本
fluCapitalization: 流通股本
- **SingleStockInfoPO:**
name: 股票名称
code: 股票代码
date: 股票信息时间
open: 开盘价格
high: 当天的最高价格
low: 当天的最低价格
volume: 当天的交易量
close: 收盘指数
subClose: 前复权收盘价格
aftClose: 后复权收盘价格
lclose: 昨日收盘价格
lsubClose: 昨日前复权收盘价格
laftClose: 昨日后复权收盘价格
rate: 未复权涨跌幅
subrate: 前复权涨跌幅
afrate: 后复权涨跌幅
aftavg_5: 后复权5日均线
- **StockSetInfoPO:**
date: 日期
List< String> haltList: 停牌股票的股票代码列表
List< String> haltNameList: 股票池中股票当天的信息

- List< SingleStockInfoPO> stockList:
- haltNumber: 停牌股票的数量
- allhalt: 股票是否全停牌, 全停牌则为true, 否则为false
- **StrategyInfoPO:**
 - strategTypeNname:
 - publicorprivate: 公有 (私有)
 - List< Integer> parameter:
 - purchasenum: 购买数量
 - tiaocangqi: 调仓期
 - tiaocangjiage: 调仓价格
- **StrategyShowPO:**
 - creatorName: 创建者姓名
 - strategyName: 策略名称
 - List< LocalDate> TimeList: 日期
 - List< Double> BasicReturn: 基础收益率
 - List< Double> StrategyReturn: 策略收益率
 - Alpha: α
 - Beta: β
 - Sharp: 夏普比率
 - zuidahuiche: 最大回撤
 - StrategyYearReturn: 策略年化收益率
 - jizhunYearReturn: 基准年化收益率

Exception定义

- **BeginInvalidException:** 用来表示起始日期大于最大有效日期
- **DateInvalidException:** 用来表示该日期段内没有股票数据
- **DateOverException:** 日期溢出时抛出该错误
 - cal:
 - last:
- **EndInvalidException:** 用来表示结束日期小于最小有效日期
- **InternetdisconnectException:** 用来表示网络连接断开或不稳定
- **LogErrorException:** 用来表示登录的用户名或密码有误
- **NoneMatchedMarketException:** 没有匹配该股票编号的市场
- **NullDateException:** 不存在该日期
- **NullMarketException:** 今日市场无数据
- **StrageNullSetExceptionntyException:** 不存在该股票池
- **NullSetInfoException:** 股票池某日期没有数据
 - set:
 - cal:
 - sdf: 日期格式 (yyyy-MM-dd)
- **NullStockIDException:** 不存在该股票时
- **PriceTypeException:** 调仓价格种类错误
- **StockHaltingException:** 股票停牌 (数据层不判断是否是星期天)
 - calendar: 停牌日期,2017-03-28则为20170328

stock: 停牌股票编号

- **StockNameRepeatException:** 在股票池中重复添加某支已存在股票
- **StockSetNameRepeatException:** 该股票池名字已存在
- **StragetyException:**
- **StrategyRepeatException:** 策略名称已存在
- **TimeShraingLackException:** 缺少该天分时信息
- **UserNameRepeatException:** 用来表示注册的用户名已经存在