stocks analysis system

AnyQuant股票策略系统

项目设计文档

团队：GreatW

团队成员：尹子越，张晨剑，张斯栋，张馨中

DATE \@ "y年M月d日" 2016年3月5日

版本：V1.0

更新历史

| 变更者 | 变更事项 |
| --- | --- |
| 尹子越 | 创建文档，基础设计 |
| 张晨剑 | 更新接口 |
| 张晨剑 | 新增接口 |

目录

TOC \t "副小标题, 1,小标题, 2,小标题 2, 3"

更新历史 PAGEREF \_Toc \h 2

目录 PAGEREF \_Toc1 \h 2

一、引言 PAGEREF \_Toc2 \h 4

1.1编写目的 PAGEREF \_Toc3 \h 4

1.2对象与范围 PAGEREF \_Toc4 \h 4

1.3参考文献 PAGEREF \_Toc5 \h 4

1.4名词与术语 PAGEREF \_Toc6 \h 4

二、系统的分层架构 PAGEREF \_Toc7 \h 4

1.系统划分为以上3个逻辑层次： PAGEREF \_Toc8 \h 5

2.系统的架构设计如下： PAGEREF \_Toc9 \h 5

3.组合视角 PAGEREF \_Toc10 \h 6

4.1开发包图 PAGEREF \_Toc11 \h 6

4.2运行时进程 PAGEREF \_Toc12 \h 6

4.3物理部署 PAGEREF \_Toc13 \h 6

5.接口视角 PAGEREF \_Toc14 \h 6

5.1模块的职责 PAGEREF \_Toc15 \h 6

5.2用户界面层的分解 PAGEREF \_Toc16 \h 7

5.3业务逻辑层的分解 PAGEREF \_Toc17 \h 7

5.4数据层的分解 PAGEREF \_Toc18 \h 14

6.信息视角 PAGEREF \_Toc19 \h 14

6.1数据持久化对象 PAGEREF \_Toc20 \h 14

6.2txt持久化格式 PAGEREF \_Toc21 \h 14

6.3数据库表 PAGEREF \_Toc22 \h 14

一、引言

1.1编写目的

本文档提供AnyQuant股票策略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。

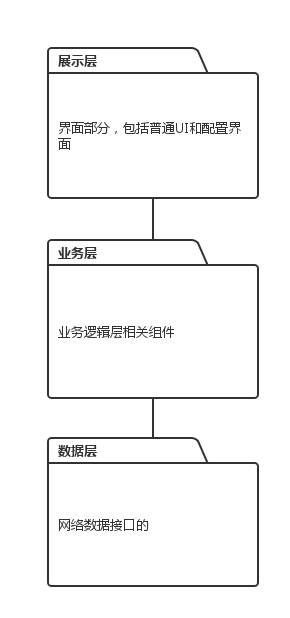
1.2对象与范围

本文档的读者是AnyQuant股票策略系统团队内部的开发和管理人员，参考了RUP的《软件架构文档模板》，用于指导下一循环的代码开发和测试工作。

1.3参考文献

《软件需求规格说明书》

1.4名词与术语

二、系统的分层架构

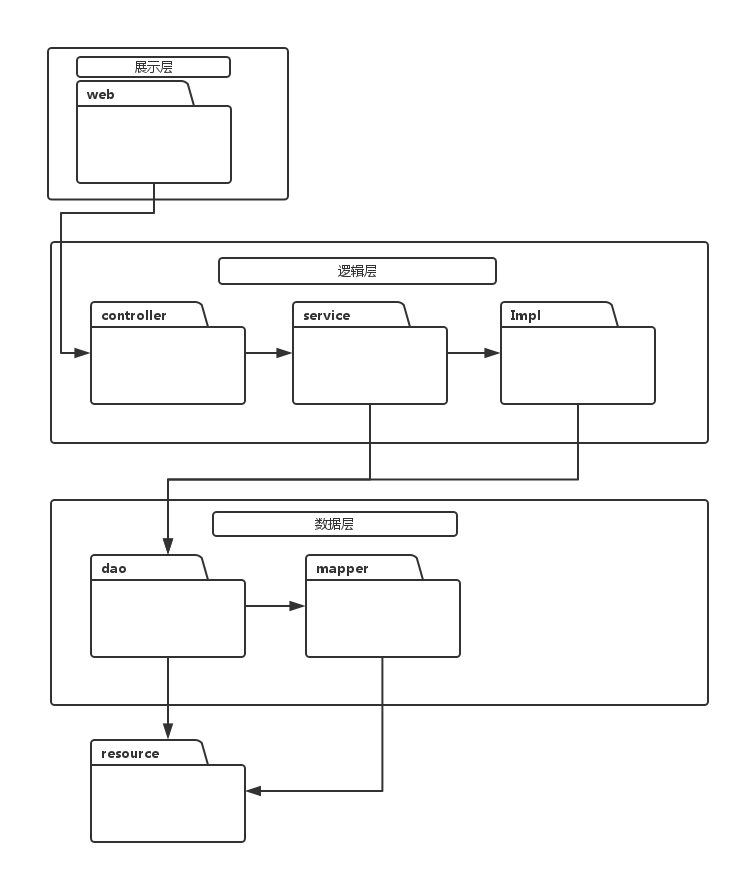
1.系统划分为以上3个逻辑层次：

1）表示层：用于前台界面展示和配置的层次。

2）业务层：包含业务控制和逻辑的层次。

3）数据层：定义和接收系统中相关数据层次。

2.系统的架构设计如下：



1）web对象，负责处理系统数据的展现和用户的交互。

2）controller对象，负责与前段交互，接受并处理http请求。

3）service对象，定义controller调用的接口

4）impl对象，负责是实现在service中定义的接口。

5）dao对象，定义数据层和逻辑层的接口。

6）mapper对象，负责调用远程数据库，实现dao内接口。

7）resource对象，远程数据库。

3.组合视角

4.1开发包图

AnyQuant股票策略系统的最终开发包设计如表4.1-1

| 开发包图（物理） | 依赖的其他开发包 |
| --- | --- |
| web | 设计图以及组长的脑洞 |
| controller | service，springMVC，impl |
| service | spring，springMVC |
| impl | spring，springMVC，service |
| dao | mapper，mybetis |
| mapper | mybetis |
| resource |  |

表4.1-1 AnyQuant股票策略系统的最终开发包设计

4.2运行时进程

在AnyQuant股票策略系统中，客户使用浏览器访问网站。

4.3物理部署

网络情况正常

5.接口视角

5.1模块的职责

视图如5.1-1所示。



图5.1－1客户端模块视图

表5.1-1客户端各层的职责

| 层 | 职责 |
| --- | --- |
| web前端 | 展示数据、和用户进行交互 |
| controller | 负责接收http请求、并处理请求返回数据 |
| 业务逻辑层 | 接受controller的请求，将从数据层获取的数据返回 |
| 数据层 | 负责接收逻辑层的请求，从数据库获取数据 |

5.2用户界面层的分解

5.3业务逻辑层的分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。

controller与业务逻辑层接口

| 接口**ID** | 连接组件 | 接口信息 | |
| --- | --- | --- | --- |
| I1 | 连接controller与Service | 语法 | DetailStrategyVO getDetailStrateg(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 股票详细策略 |
| 不变量 | 分析日期 |
| I2 | 连接controller与Service | 语法 | ArrayList<FavouriteStock> getFavourite(ArrayList<String> idlist) |
| 前置条件 | 输入用户收藏列表 |
| 后置条件 | 返回收藏股票的信息 |
| 不变量 | 日期区间 |
| I3 | 连接controller与Service | 语法 | ForecastData getForecastData(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回走势预测 |
| 不变量 | 预测区间 |
| I4 | 连接controller与Service | 语法 | ArrayList<PyTradeData> getPyTrade(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回走势预测 |
| 不变量 | 预测区间 |
| I5 | 连接controller与Service | 语法 | StockGradeVO getCurrentInfo(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取股票评分 |
| 不变量 | 时间区间 |
| I6 | 连接controller与Service | 语法 | ArrayList<Industry> getIndustryAnalysis() |
| 前置条件 | 无 |
| 后置条件 | 获取全行业分析 |
| 不变量 | 时间区间 |
| I7 | 连接controller与Service | 语法 | StockInsTextVO getStockAnalysis(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取实时分析 |
| 不变量 | 无 |
| I8 | 连接controller与Service | 语法 | StockInsModelVO getModelAnalysis(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取大盘相关性 |
| 不变量 | 计算时间区间 |
| I9 | 连接controller与Service | 语法 | StockRelativeData getRelativeData(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取大盘相关性 |
| 不变量 | 输入股票信息 |
| I10 | 连接controller与Service | 语法 | ArrayList<SingleInfo> getSingleInfo() |
| 前置条件 | 无 |
| 后置条件 | 获取所有股票列表 |
| 不变量 | 无 |
| I11 | 连接controller与Service | 语法 | StockComparison getStockComparsion(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回股票对比所需信息 |
| 不变量 | 对比时间区间 |
| I12 | 连接controller与Service | 语法 | StockInfo getStockInfo(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回公司信息 |
| 不变量 | 无 |
| I13 | 连接controller与Service | 语法 | StockSeason getStockSeason(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取公司季度报表 |
| 不变量 | 无 |
| I14 | 连接controller与Service | 语法 | ArrayList<TabTablesData> getTablesInfo(String id, LocalDate begin, LocalDate end) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取股票详细信息 |
| 不变量 | 无 |
| I15 | 连接controller与Service | 语法 | ArrayList<TabTableInsVO> getTablesInfo(String id, LocalDate begin, LocalDate end) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取股票详细信息及相关分析 |
| 不变量 | 无 |

业务逻辑层与数据层层接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口**ID** | 连接组件 | 接口信息 |  |
| I8 | 连接Service与dao | 语法 | ArrayList<Industry> getIndustryAssess() |
| 前置条件 | 无 |
| 后置条件 | 获取全行业分析 |
| 不变量 | 时间区间 |
| I9 | 连接Service与dao | 语法 | ArrayList<StockGradeVO> getStockGrade(String industryid) |
| 前置条件 | 输入行业信息 |
| 后置条件 | 返回行业内股票评分结果 |
| 不变量 | 评分时间区间 |
| I10 | 连接Service与dao | 语法 | ArrayList<IndustryInnerRank> getTurnoverRateRank(String industryid) |
| 前置条件 | 输入行业信息 |
| 后置条件 | 返回行业内个项评分排名 |
| 不变量 | 评分时间区间 |
| I11 | 连接Service与dao | 语法 | ArrayList<IndustryInnerRank> getDeviationRank(String industryid) |
| 前置条件 | 输入行业信息 |
| 后置条件 | 返回行业内个项评分排名 |
| 不变量 | 评分时间区间 |
| I12 | 连接Service与dao | 语法 | ArrayList<IndustryInnerRank> getPriceRank(String industryid) |
| 前置条件 | 输入行业信息 |
| 后置条件 | 返回行业内个项评分排名 |
| 不变量 | 评分时间区间 |
| I13 | 连接Service与dao | 语法 | ArrayList<SingleInfo> getSingleInfo() |
| 前置条件 | 无 |
| 后置条件 | 返回所有股票简要信息 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<TabTablesData> getTabTablesData(String id, String start , String end) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回股票详细信息 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<TabTablesData> getBenchTabtable(String id ,String start ,String end) |
| 前置条件 | 输入大盘信息 |
| 后置条件 | 返回大盘详细信息 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<CandleData> getCandleData(String id, String start , String end) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 返回股票k线图数据 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<ForecastData> getForecastData(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取预测信息 |
| 不变量 | 预测时间区间 |
| I14 | 连接Service与dao | 语法 | PytradeDataPo getPyTrade(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取预测信息 |
| 不变量 | 预测时间区间 |
| I14 | 连接Service与dao | 语法 | StockRelativeData getRelativeData(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取大盘相关性 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<FavouriteStock> getFavourite(Map map) |
| 前置条件 | 输入收藏股票信息 |
| 后置条件 | 获取收藏股票的信息 |
| 不变量 | 时间区间 |
| I14 | 连接Service与dao | 语法 | ArrayList<FavouriteStock> getBenchFavourite(String start ,String end) |
| 前置条件 | 输入大盘信息 |
| 后置条件 | 获取收藏界面所需大盘信息 |
| 不变量 | 时间区间 |
| I14 | 连接Service与dao | 语法 | StockGradeVO getStockGrade(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取个股评分 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | ArrayList<GradeStatistics> getStatistic(String id , String start,String end) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取评分原始数据 |
| 不变量 | 数据时间区间 |
| I14 | 连接Service与dao | 语法 | StockInfo getStockInfo(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取公司信息 |
| 不变量 | 无 |
| I14 | 连接Service与dao | 语法 | StockSeason getStockSeason(String id) |
| 前置条件 | 输入股票信息 |
| 后置条件 | 获取公司季度信息 |
| 不变量 | 无 |

5.4数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对网络的http请求和数据的缓存。由于持久化数据的保存存在多种形式：数据库、http请求json文件，所以抽象了数据服务。

6.信息视角

6.1数据持久化对象

系统的Statistics类就是对应的相关的实体类，在此只做简单的介绍。如表6.1所示：

表6.1 Statistics类简单介绍

| 类名 | 包含的属性 |
| --- | --- |
| BenchStatistics | 开盘价，最高价，最低价，收盘价，成交量，后复权价 |
| StockStatistics | 开盘价，最高价，最低价，收盘价，成交量，后复权价，换手率，市盈率，市净率 |

6.2txt持久化格式

使用Serializable将数据类序列化保存到txt文件。

6.3数据库表

|  |
| --- |
| 表名 |
| Stock\_2010 |
| Stock\_2011 |
| Stock\_2012 |
| Stock\_2013 |
| Stock\_2014 |
| Stock\_2015 |
| Stock\_2016 |
| bench |
| bp\_predict |
| industry\_tag |
| predict |
| relative |
| season |
| short\_ticks |
| stock\_evaluate |
| stock\_info |