

**AnyQuant股票分析系统**

**软件架构设计文档**

**今天下午吃什么队**

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 王志豪 | 2016/2/29 | 初始版本 | V1.0修正版 |
| 孙晓晨 | 2016/3/28 | 二代版本 | V2.0修正版 |

**目录**

1. **引言**

**1.1、目的**

**1.2、范围**

**1.3、参考文献**

**1.4、名词与术语**

1. **逻辑视角**

**2.1分层架构图**

**2.2逻辑架构图**

1. **物理视角**

**3.1物理包划分**

**3.2物理包图**

1. **接口视角**

**4.1模块的职责**

**4.2模块的接口规范**

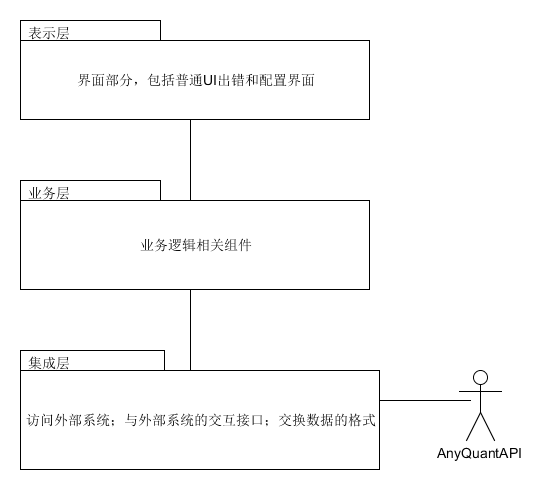
1. **信息视角**

1、引言

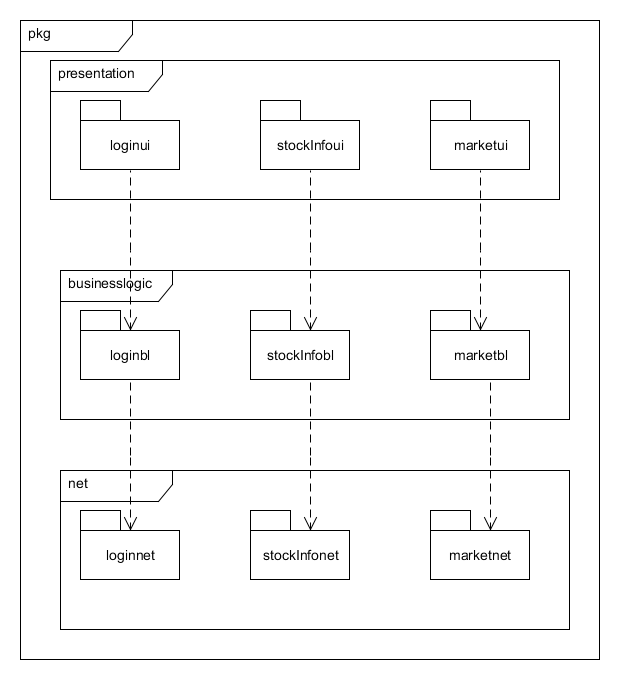
|  |  |
| --- | --- |
| 编写目的 | 本文档提供股票信息分析系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。 |
| 对象与范围 | 本文档的读者是”今天下午吃什么”团队内部的开发和管理人员,参考了《软件工程与计算（三）》中的《Sharing Team软件架构设计文档片段》，用于指导下一循环的代码开发和测试工作。 |
| 参考文献 | 《项目启动文档》  《软件需求规格说明书》  《软件架构文档模板》  《软件工程与计算（卷三）》 |
| 名词与术语 |  |

1. 逻辑视角

2.1、分层架构图



2.2、逻辑包图

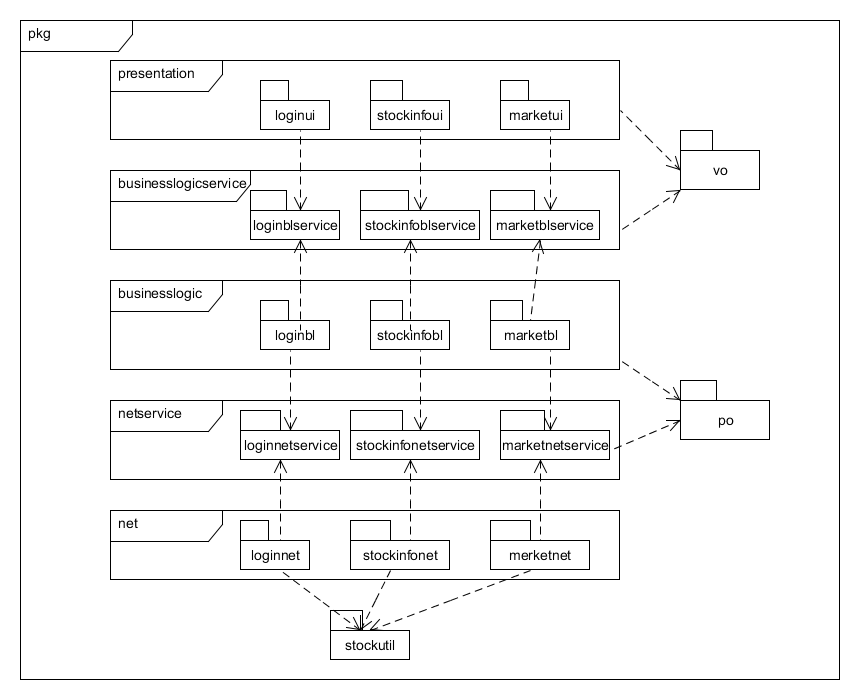


1. 组合视角

3.1、物理包的划分

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| loginui | vo、loginblservice、stockinfoui、marketui、config |
| loginblservice | Vo |
| loginbl | Loginnetservice、config、vo |
| loginnetservice | po |
| loginnet | po、config、stockutil |
| stockinfoui | Vo、stockinfoblservice、config |
| stockinfoblservice | vo |
| stockinfobl | Stockinfonetservice、config、vo |
| stockinfonetservice | po |
| stockinfonet | po、config、stockutil |
| marketui | vo、marketblservice、config |
| marketblservice | vo |
| marketbl | Marketnetservice、config、vo |
| marketnetservice | po |
| marketnet | Po、config、stockinfoutil |
| vo |  |
| po |  |
| config |  |
| stockutil |  |

3.2、物理包图



1. 接口视角

## 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。

|  |
| --- |
| 启动模块 |

|  |
| --- |
| 展示层 |

|  |
| --- |
| 业务逻辑层 |

|  |
| --- |
| 网络模块 |

**图 客户端模块视图**

**表2 客服端各层的职责**

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的股票信息分析系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理 |
| 客户端网络模块 | 通过发送HTTP请求获取服务器上的远程数据1 |

**层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| LoginBLService  StockInfoBLService  MarketBLService | 客户端展示层 | sd客户端业务逻辑层 |
| LoginNetService  StockInfoNetService  MarketNetService | 客户端业务逻辑层 | 服客户端网络层 |

### 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 |
| MainFrame | 界面Frame,负责界面的显示和界面的跳转 |

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | Init(String[] args) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame以及LoginPanel |

用户界面层需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| VisualizationBLService  StockInfoBLService  MarketBLService | 获取可视化图表的的逻辑接口  获取股票列表及其详细信息的逻辑接口  获取大盘信息的逻辑接口 |

### 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** |
| VisualizationBLService  StockInfoBLService  MarketBLService | 获取可视化图表  获取股票列表及其详细信息  获取大盘信息 |

### 4.2.2.2业务逻辑层模块的接口规范

**VisualizationBL层接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| VisualizationMarketData.getKlineDate | 语法 | | public List<KlineVO> getKlineDate(String start, String end, String marketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得大盘K线图信息 |
| VisualizationMarketData. getLineAndHistogramStockDate | 语法 | | public List<LineAndHistogramChartVO> getLineAndHistogramStockDate(String start, String end, String marketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得大盘历史信息 |
| VisualizationMarketPanel. getKlineStockPanel | 语法 | | public JPanel getKlineStockPanel (String start, String end, String markketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得可视化大盘K线图 |
| VisualizationMarketPanel. getLineAndHistogramStockPanel | 语法 | | public JPanel getLineAndHistogramStockPanel(String start,String end,String markketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得可视化大盘历史信息 |
| VisualizationRecommendPanel. getScatterPanel | 语法 | | public JPanel getScatterPanel(); |
|  | 前置条件 | | 无 |
|  | 后置条件 | | 获得可视化相关行业推荐信息图 |
| VisualizationStockData.getKlineDate | 语法 | | public List<KlineVO> getKlineDate(String start, String end, String marketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得股票K线图信息 |
| VisualizationStockData. getLineAndHistogramStockDate | 语法 | | public List<LineAndHistogramChartVO> getLineAndHistogramStockDate(String start, String end, String marketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得股票历史信息 |
| VisualizationStockPanel. getKlineStockPanel | 语法 | | public JPanel getKlineStockPanel (String start, String end, String markketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得可视化股票K线图 |
| VisualizationStockPanel. getLineAndHistogramStockPanel | 语法 | | public JPanel getLineAndHistogramStockPanel(String start,String end,String markketNum); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 获得可视化股票历史信息 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| LoginNetService.connect | | 检验和AnyQuantity服务器的连接 | |

**StockInfoBL层接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| StockInfoBLService.getAll | 语法 | | Public ArrayList<String> getAll() |
|  | 前置条件 | | 无 |
|  | 后置条件 | | 返回所有可用的股票列表。 |
| StockInfoBLService.getStockByYearOrByPlace | 语法 | | Public ArrayList<String> getStockByYearOrByPlace(String year,String place); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 返回指定年份和交易所的可用股票列表 |
| StockInfoBLService.getStockInfo | 语法 | | Public ArrayList<StockInfoVo> getStockInfo(String stock); |
|  | 前置条件 | | 选择某只股票 |
|  | 后置条件 | | 返回过去一个月的全部交易数据 |
| StockInfoBLService.getStockInfoByTime | 语法 | | Public ArrayList<StockInfoVO> getStockInfoByTime(String stock,String startTime,String endTime); |
|  | 前置条件 | | startTime和endTime不能为空且格式符合要求 |
|  | 后置条件 | | 返回指定时间内的交易数据 |
| StockInfoBLService.getStockInfoBySelect | 语法 | | public ArrayList<StockInfoVO> getStockInfoBySelect(String stock, String startTime,String endTime,SelectValueBean open,SelectValueBean high,SelectValueBean low, SelectValueBean Close , SelectValueBean adj\_price,SelectValueBean volume,SelectValueBean pe,SelectValueBean pb); |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 返回筛选条件下的股票列表 |
| StockInfoBLService. getHistoryInfo | 语法 | | public ArrayList<HistoryInfoVO> getHistoryInfo(String codeNum); |
|  | 前置条件 | | 选中某只股票 |
|  | 后置条件 | | 返回股票历史信息 |
| StockInfoBLService. getNowTimeStockInfo | 语法 | | public NowTimeStockInfoVO getNowTimeStockInfo(String codeNum); |
|  | 前置条件 | | 选中某只股票 |
|  | 后置条件 | | 返回股票实时信息 |
| StockInfoBLService. getMACDData | 语法 | | public ArrayList<macdVO> getMACDData(String startTime, String endTime, String codeNum, String type); |
|  | 前置条件 | | 选中某只股票 |
|  | 后置条件 | | 返回该股票的MACD数据 |
| StockInfoBLService.update | 语法 | | public void update(); |
|  | 前置条件 | | 无 |
|  | 后置条件 | | 更新股票信息 |
| StockInfoBLService. getStockInscreasment | 语法 | | public ArrayList<IncreasementVO> getStockInscreasment(String startTime,String endTime,String codeNum); |
|  | 前置条件 | | 选中某只股票 |
|  | 后置条件 | | 返回股票涨幅 |
| StockInfoBLService. getMA | 语法 | | public ArrayList<MAVO> getMA(int days,String startTime,String endTime,String codeNum); |
|  | 前置条件 | | 选中某只股票 |
|  | 后置条件 | | 返回股票MA数据 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| StockInfoNetService.getAll | | 返回所有可用的股票列表 | |
| StockInfoNetService.getStockByYearOrByPlace | | 返回指定年份和交易所的可用股票列表 | |
| StockInfoNetService.getStockInfo | | 返回过去一个月的全部交易数据 | |
| StockInfoNetService.getStockInfoByTime | | 返回指定时间内的交易数据 | |

**MarketBL层接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| MarketBLService.getAllBenchMark | 语法 | | Public ArrayList<String> getAllBenchMark() |
|  | 前置条件 | | 无 |
|  | 后置条件 | | 返回所有可用大盘。 |
| MarketBLService. getBenchMarkInfoByTimeOrBySelect | 语法 | | public ArrayList<BenchMarkInfoVO> getBenchMarkInfoByTimeOrBySelect(String startTime, String endTime, String benchmark ,SelectValueBean open, SelectValueBean high ,SelectValueBean close,SelectValueBean low ,SelectValueBean volume, SelectValueBean adj\_price) |
|  | 前置条件 | | 可以为空 |
|  | 后置条件 | | 返回筛选条件下的大盘信息列表 |
| MarketBLService. getHistoryInfo | 语法 | | public ArrayList<HistoryInfoVO> getHistoryInfo(String benchmark); |
|  | 前置条件 | | 选中某大盘 |
|  | 后置条件 | | 返回大盘历史信息 |
| MarketBLService. getNowTimeBenchMarkInfo | 语法 | | public NowTimeBenchMarkInfoVO getNowTimeBenchMarkInfo(); |
|  | 前置条件 | | 无 |
|  | 后置条件 | | 返回大盘实时信息 |
| MarketBLService. getQuadrantDiagramInfo | 语法 | | public ArrayList<QuadrantDiagramVO> getQuadrantDiagramInfo(String startDate,int days); |
|  | 前置条件 | | 时间不为空 |
|  | 后置条件 | | 返回象限图数据 |
| MarketBLService. getMartketInscreasment | 语法 | | public ArrayList<IncreasementVO> getMartketInscreasment(String startTime,String endTime,String marketName); |
|  | 前置条件 | | 选中某大盘 |
|  | 后置条件 | | 返回大盘涨幅 |
| MarketBLService. getMA | 语法 | | public ArrayList<MAVO> getMA(int days,String startTime,String endTime,String marketName); |
|  | 前置条件 | | 选中某大盘 |
|  | 后置条件 | | 返回大盘MA数据 |
| MarketBLService.geBenchMarkInfoByTimeOrByMarket | 语法 | | Public ArrayList<BenchMarkInfoVO> getBenchMarkInfoByTimeOrByMarket(String startTime,String endTime,String benchmark); |
|  | 前置条件 | | startTime和endTime不能为空且格式符合要求,benchmark不能为空 |
|  | 后置条件 | | 返回指定时间内指定大盘的数据 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| MarketBLService.getAllBenchMark | | 返回所有可用大盘。 | |
| MarketBLService.getBenchMarkInfoByTimeOrByMarket | | 返回指定时间内指定大盘的数据 | |

## 4.2.3网络层的分解

### 4.2.3.1网络层模块的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| LoginNetService  StockInfoNetService  MarketNetService | 网络连接检查  从AnyQuant获取股票列表及其详细信息  从AnyQuant获取大盘信息 |

### 4.2.3.2网络层模块的接口规范

**LoginNet层接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| LoginNetService.connect | 语法 | Public boolean connect(); |
|  | 前置条件 | 无 |
|  | 后置条件 | 如果可以正常访问到AnyQuant服务器则返回true，否则返回false |

**StockInfoNet层接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StockInfoNetService.getAll | 语法 | Public ArrayList<String> getAll() |
|  | 前置条件 | 无 |
|  | 后置条件 | 返回所有可用的股票列表。 |
| StockInfoNetService.getStockByYearOrByPlace | 语法 | Public ArrayList<String> getStockByYearOrByPlace(String year,String place); |
|  | 前置条件 | 可以为空 |
|  | 后置条件 | 返回指定年份和交易所的可用股票列表 |
| StockInfoNetService.getStockInfo | 语法 | Public ArrayList<StockInfoPo> getStockInfo(String stock); |
|  | 前置条件 | 选择某只股票 |
|  | 后置条件 | 返回过去一个月的全部交易数据 |
| StockInfoNetService.getStockInfoByTime | 语法 | Public ArrayList<StockInfoPO> getStockInfo(String stock,String startTime,String endTime); |
|  | 前置条件 | startTime和endTime不能为空且格式符合要求 |
|  | 后置条件 | 返回指定时间内的交易数据 |

**MarketNet层接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| MarketNetService.getAllBenchMark | 语法 | Public ArrayList<String> getAllBenchMark() |
|  | 前置条件 | 无 |
|  | 后置条件 | 返回所有可用大盘。 |
| MarketNetService.getBenchMarkInfoByTimeOrByMarket | 语法 | Public ArrayList<BenchMarkInfoPO> getBenchMarkInfoByTimeOrByMarket(String startTime,String endTime,String benchmark); |
|  | 前置条件 | startTime和endTime不能为空且格式符合要求,benchmark不能为空 |
|  | 后置条件 | 返回指定时间内指定大盘的数据 |

1. 信息视角

**VO定义**

StockInfoVO:name:代号、date:日期、open:开盘价、high:最高价、low:最低价、close:收盘价、adj\_price:后复权价、volume:成交量、turnover:换手率、pe:市盈率、pb:市净率

BenchMarkInfoVO:name:代号、date:日期、open:开盘价、close:收盘价、high:最高价

HistoryInfoVO: volume：成交量、price：成交价、sum：总价, date：日期, up：涨跌)

**PO定义**

StockInfoPO:name:代号、date:日期、open:开盘价、high:最高价、low:最低价、close:收盘价、adj\_price:后复权价、volume:成交量、turnover:换手率、pe:市盈率、pb:市净率

BenchMarkInfoPO:name:代号、date:日期、open:开盘价、close:收盘价、high:最高价

HistoryInfoPO: volume：成交量、price：成交价、sum：总价, date：日期, up：涨跌)