STOck Analysis system

AnyQuant股票策略系统

软件测试文档

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2月29日

版本： V1.0

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 人员 | 时间 | 更新原因 | 版本号 |
| 张晨剑 | 2016-03-07 | 创建文档 | v1.0.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

迭代一单元测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant1-001 | |
| **测试单元** | | | QueryStatisticsImpl | |
| **是否有测试代码** | | 是 | **测试代码** | QueryStatisticServiceTest |
| **前提条件** | 展示层调用查询股票，以及大盘数据的方法，逻辑层准备向数据层请求 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh300,2015-12-4,2015-12-5”，以及“sh300”为参数，分别调用queryBenchmarkStatistics(String id,Date start,Date end),queryBenchmarkStatistics(String id)比较结果 | 相同 | 相同 | Pass |
| 2 | 以“sh600000”为参数调用getStock（String id）方法，比较得到的stock对象的id | 所得stock对象的id为sh600000 | 控制台显示json error | fail |
| 3 | 以“sh600000,2016-2-4,2016-3-5”为参数调用getStock（String id,Date start,Date end）方法，比较得到的stock对象的时间 | 时间为2016-02－01 | 时间为2016-02－01 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-3-6 | |
| **测试用例实际执行成本** | | 0.15人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant1-002 | |
| **测试单元** | | | FavouriteStockMgtImpl | |
| **是否有测试代码** | | 是 | **测试代码** | FavouriteStockServiceTest |
| **前提条件** | 用户设置收藏股票，新增或者删除 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getFavouriteStock()方法，得到第一个股票编号，与txt文件中对照 | sh600002 | sh600002 | Pass |
| 2 | 以“sh600000”为参数调用setFavouriteStock（String id）方法，查看txt | txt文件中新增sh600000股票 | txt文件中新增sh600000股票 | Pass |
| 3 | 以“sh600000”为参数调用deleteFavouriteStock（String id）方法，查看txt | txt文件中删除sh600000股票 | txt文件中删除sh600000股票 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-3-7 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant1-003 | |
| **测试单元** | | | FavouriteStockService | |
| **是否有测试代码** | | 是 | **测试代码** | FileFavouriteStock |
| **前提条件** | 加载首界面，显示收藏股票 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getFavouriteStock()方法，得到股票APIUrl列表 | 列表存在并且非空 | 列表存在并且非空 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-3-7 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant1-004 | |
| **测试单元** | | | FileStockNameComplement | |
| **是否有测试代码** | | 是 | **测试代码** | NameComplementService |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh600000”为参数调用stockNameComplement(String id)方法 | 浦发银行 | 浦发银行 | Pass |
| 2 | 以“sz000012”为参数调用stockNameComplement(String id)方法 | 南玻A | 南玻A | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-3-6 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant1-005 | |
| **测试单元** | | | HttpBasedUrl | |
| **是否有测试代码** | | 是 | **测试代码** | URLProvideServiceTest |
| **前提条件** | 数据层请求获得URl | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getStockList()方法，得到股票APIUrl列表 | 列表存在并且非空 | 列表存在并且非空 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-3-6 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

迭代一单元测试结论

单元测试第一次通过率为90%，发现一个缺陷，问题在于json语句有一处编码错误，修正缺陷后，单元测试全部通过。

迭代一集成测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 1 | 涉及模块 | presentation,ConversionBl,DataProvide | | |
| 依赖条件 | 网络状况良好，API数据足够，输入正确 | | | | |
| 步骤 | 1.点击搜索框  2.键入所要查询股票的编号  3.选择搜索结果 | | | | |
| 预期结果 | 显示该股票的近一个月内相关信息 | | | | |
| 实际结果 | 显示该股票的近一个月内相关信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 2 | 涉及模块 | presentation,ConversionBl,DataProvide | | |
| 依赖条件 | 网络状况良好，API数据足够，输入正确 | | | | |
| 步骤 | 1.点击搜索框  2.键入所要查询股票的编号，并输入起止日期  3.选择搜索结果 | | | | |
| 预期结果 | 显示该股票的指定日期内相关信息 | | | | |
| 实际结果 | 显示该股票的指定日期内相关信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 3 | 涉及模块 | presentation,ConversionBl,DataProvide | | |
| 依赖条件 | 网络状况良好，API数据足够，输入正确 | | | | |
| 步骤 | 1.点击搜索框  2.键入大盘编号  3.选择搜索结果 | | | | |
| 预期结果 | 显示大盘一个月内的相关信息 | | | | |
| 实际结果 | 显示大盘一个月内的相关信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 4 | 涉及模块 | presentation,ConversionBl,DataProvide | | |
| 依赖条件 | 网络状况良好，API数据足够，输入正确 | | | | |
| 步骤 | 1.点击搜索  2.键入大盘编号，并输入起止时间  3.选择搜索结果 | | | | |
| 预期结果 | 显示大盘起止时间内的相关信息 | | | | |
| 实际结果 | 显示大盘起止时间内的相关信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 5 | 涉及模块 | presentation,FavouriteBl,DataFavourite | | |
| 依赖条件 | 用户使用过该软件并且收藏过股票，本地文件保存了记录 | | | | |
| 步骤 | 1.打开软件 | | | | |
| 预期结果 | 显示用户收藏的股票信息 | | | | |
| 实际结果 | 显示用户收藏的股票信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 6 | 涉及模块 | presentation,FavouriteBl,DataFavourite | | |
| 依赖条件 | 本地存在保存信息的文件 | | | | |
| 步骤 | 1.查看股票列表  2.收藏股票 | | | | |
| 预期结果 | 将股票添加到用户收藏列表 | | | | |
| 实际结果 | 将股票添加到用户收藏列表 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 7 | 涉及模块 | presentation,FavouriteBl,DataFavourite | | |
| 依赖条件 | 本地存在保存信息的文件 | | | | |
| 步骤 | 1.查看股票列表  2.取消收藏股票 | | | | |
| 预期结果 | 将股票从用户收藏列表中删除 | | | | |
| 实际结果 | 将股票从用户收藏列表中删除 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

迭代一集成测试结论

集成测试通过率为100%，经分析，由于单元测试的存在减少了编码错误，提高了集成测试的通过率。迭代一功能需求较为简单也在一定程度上提高了集成测试的正确率。

界面层初步完成，需要经过大量的优化。优化后，原先的集成测试同样可以通过，保持高通过率。

迭代二单元测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-001 | |
| **测试单元** | | | ComplementNameImpl | |
| **是否有测试代码** | | 是 | **测试代码** | ComplementNameImpl |
| **前提条件** | 输入正确股票以及大盘信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以Industry.BLHY为参数，调用ComplementNameImpl.IndustryComplement，输出结果 | 玻璃行业 | 玻璃行业 | Pass |
| 2 | 以“sh600000”为参数，调用ComplementNameImpl.StockIdComplement方法，输出结果 | 浦发银行 | 浦发银行 | pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-002 | |
| **测试单元** | | | FuzzyQueryImpl | |
| **是否有测试代码** | | 是 | **测试代码** | FuzzyQueryServiceTest |
| **前提条件** | 网络连接，能够获取股票列表 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh6000”为参数，调用FuzzyQueryImpl.FuzzyQueryString，输出结果 | 以sh6000开头的10条股票 | 以sh6000开头的10条股票 | Pass |
| 2 | 以“sh6000”为参数，调用FuzzyQueryImpl.FuzzyQuery，输出所得股票列表的id | 以sh6000开头的10条股票 | NUllpointer异常 | fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-003 | |
| **测试单元** | | | QueryAnyQuantImpl | |
| **是否有测试代码** | | 是 | **测试代码** | QueryStaticsServiceTest |
| **前提条件** | 输入正确股票以及大盘信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh300,2015-12-4,2015-12-5”，以及“sh300”为参数，分别调用queryBenchmarkStatistics(String id,Date start,Date end),queryBenchmarkStatistics(String id)比较结果 | 相同 | 相同 | Pass |
| 2 | 以“sh600000”为参数调用getStock（String id）方法，比较得到的stock对象的id | 所得stock对象的id为sh600000 | 所得stock对象的id为sh600000 | Pass |
| 3 | 以“sh600000,2016-2-4,2016-3-5”为参数调用getStock（String id,Date start,Date end）方法，比较得到的stock对象的时间 | 时间为2016-02－01 | 时间为2016-02－01 | Pass |
| 4 | 以“sh600000,2016-2-4,2016-3-5”为参数调用getWeekKLineString id,Date start,Date end）方法，显示返回对象的时间 | 均为区间内周一日期 | 返回空对象，没有结果输出 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-004 | |
| **测试单元** | | | AssessEnergyImpl | |
| **是否有测试代码** | | 是 | **测试代码** | AssessEnergyServiceTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用搜索接口搜出sh600000一个月内的数据，以此为参数调用AssessEnergy（StockAnyQuant stock）方法 | 0-100的数字 | 0-100的数字 | Pass |
| 2 | 调用搜索接口搜出sh600000一个月内的数据，以此为参数调用AssessTurnoverRate(StockAnyQuant stock）方法 | 0-100的数字 | 0-100的数字 | pass |
| 3 | 调用搜索接口搜出sh600000一个月内的数据，以此为参数调用AssessTradingVolume(StockAnyQuant stock）方法 | 0-100的数字 | 0-100的数字 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-005 | |
| **测试单元** | | | AssessPBImpl | |
| **是否有测试代码** | | 是 | **测试代码** | AssessPBServiceTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用搜索接口搜出sh600000一个月内的数据，以此为参数调用AssessPB（StockAnyQuant stock）方法 | 0-100的数字 | 0-100的数字 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-006 | |
| **测试单元** | | | AssessPEImpl | |
| **是否有测试代码** | | 是 | **测试代码** | AssessPEServiceTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用搜索接口搜出sh600000一个月内的数据，以此为参数调用AssessPE（StockAnyQuant stock）方法 | 0-100的数字 | 0-100的数字 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-007 | |
| **测试单元** | | | IndustryAnalysisImpl | |
| **是否有测试代码** | | 是 | **测试代码** | IndustryAssessTest |
| **前提条件** | 输入行业信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用popularIndustry()方法，输出返回数组的前十条数据 | 行业类型 | 返回数组为空 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-10 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-011 | |
| **测试单元** | | | StockRanger | |
| **是否有测试代码** | | 是 | **测试代码** | StockRangerTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以"sh600000**"**, DateFormat.*strToDate*(“2016-04-10")为参数新建ATR对象，并输出计算结果 | 0.4166666 | 0.4166666 | Pass |
| 2 | 以"sh600000**"**, DateFormat.*strToDate*(“2016-04-12")为参数新建ATR对象，并输出计算结果 | 0.4033333 | 0.4033333 | pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-11 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-012 | |
| **测试单元** | | | AnyQuantRanger | |
| **是否有测试代码** | | 是 | **测试代码** | StatisticsRanger |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以"sh600000**"**, DateFormat.*strToDate*(“2016-04-05”)，DateFormat.*strToDate*(“2016-04-13”)为参数调用AnyQuantRanger.getOneRanger,输出期间的结果 | 10个double | 10个double | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-11 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-013 | |
| **测试单元** | | | AnyQuantSearch | |
| **是否有测试代码** | | 是 | **测试代码** | Refresh |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用AnyQuantSearch.refreshStock,输出期间的结果 | 更新txt中文件 | 更新txt中文件，但是数据有误 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-11 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant2-014 | |
| **测试单元** | | | HttpBasedCompanyDetail | |
| **是否有测试代码** | | 是 | **测试代码** | CompanyProvide |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以sh600012为参数调用HttpBasedCompanyDetail.getStock,对应日期、类型、描述 | 对应日期、类型、描述 | 对应日期、类型、描述 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-4-11 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

迭代二单元测试结论

单元测试第一次通过率为78.9%，发现4个缺陷，主要问题在于返回对象为空，或者没有返回搜索结果，主要问题在于对于股票垃圾数据以及调用的时候处理不当导致。

经过调试后，用例全部通过。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 1 | 涉及模块 | presentation,CoversionBl,DataFavourite | | |
| 依赖条件 | 输入股票信息正确 | | | | |
| 步骤 | 1.输入股票信息  2.查看结果 | | | | |
| 预期结果 | 显示模糊搜索结果 | | | | |
| 实际结果 | 显示模糊搜索结果 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

迭代二集成测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 2 | 涉及模块 | presentation,CoversionBl,Data | | |
| 依赖条件 | 输入股票信息正确 | | | | |
| 步骤 | 1.进入股票页面  2.进入股票表格 | | | | |
| 预期结果 | 显示k线图、成交量图和股票表格 | | | | |
| 实际结果 | 显示k线图、成交量图和股票表格 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 3 | 涉及模块 | presentation,CoversionBl,StockAssessBl,Data | | |
| 依赖条件 | 输入股票信息正确 | | | | |
| 步骤 | 1.进入股票页面  2.股票统计分析界面 | | | | |
| 预期结果 | 显示股票各项数据以及MACD | | | | |
| 实际结果 | 显示股票各项数据以及MACD | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 4 | 涉及模块 | presentation,CoversionBl,StockAssessBl,Data | | |
| 依赖条件 | 输入股票信息正确 | | | | |
| 步骤 | 1.添加股票进入对比页面  2.进入股票对比页面 | | | | |
| 预期结果 | 显示股票各项数据的对比 | | | | |
| 实际结果 | 显示股票各项数据的对比 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 5 | 涉及模块 | presentation,CoversionBl,StockAssessBl,Data | | |
| 依赖条件 | 行业数据持久化 | | | | |
| 步骤 | 1.进入行业界面  2.点击行业 | | | | |
| 预期结果 | 显示行业各项数据的对比以及行业内股票各项数据的对比 | | | | |
| 实际结果 | 显示行业各项数据的对比以及行业内股票各项数据的对比 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

迭代二集成测试结论

集成测试通过率为100%，经分析，由于单元测试的存在减少了编码错误，提高了集成测试的通过率。

界面层初步完成，需要经过大量的优化。优化后，原先的集成测试同样可以通过，保持高通过率。

迭代三单元测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-001 | |
| **测试单元** | | | IndustryGradeMapper | |
| **是否有测试代码** | | 是 | **测试代码** | IndustryMapperTest |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getIndustryAssess（）方法，并打印返回的结果 | 行业排名列表 | 行业排名列表 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-002 | |
| **测试单元** | | | SingleInfoMapper | |
| **是否有测试代码** | | 是 | **测试代码** | SingleInfoMapperTest |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getSingleInfo()方法，并打印返回的结果 | 所有股票信息 | 所有股票信息 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-003 | |
| **测试单元** | | | StockDataMapper | |
| **是否有测试代码** | | 是 | **测试代码** | StockDataMapperTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**“**为参数，调用getTabTable(String id,String start,String end）方法并打印结果 | sh600000的相应图表数据 | sh600000的相应图表数据 | Pass |
| 2 | 以”sh600000**“**为参数，调用getCandleData(String id,String start,String end）方法并打印结果 | sh600000的k线图数据 | sh600000的k线图数据 | Pass |
| 3 | 以”hs300**“**为参数，调用getBenchTabtable(String id,String start,String end）方法并打印结果 | 大盘相关图表数据 | 数据库异常 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-004 | |
| **测试单元** | | | ForecastDataTest | |
| **是否有测试代码** | | 是 | **测试代码** | StockDataMapperTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**“**为参数，调用getForecastData(String id）方法并打印结果 | 股票下一天价格预测区间 | 股票下一天价格预测区间 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-005 | |
| **测试单元** | | | RelativeDataTest | |
| **是否有测试代码** | | 是 | **测试代码** | StockDataMapperTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**“**为参数，调用getRelativeData(String id）方法并打印结果 | 股票与大盘的相关性系数 | 未知column异常 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-006 | |
| **测试单元** | | | StockFavouriteMapper | |
| **是否有测试代码** | | 是 | **测试代码** | StockFavouriteMapperTest |
| **前提条件** | 输入收藏股票列表 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**,“sh600004”组成的list为参数为参数，调用getFavourite(ArrayList)方法，并打印返回的结果 | sh600000,sh6000004,对应五天的数据 | sh600000,sh6000004,对应五天的数据，但缺少股票中文名 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-007 | |
| **测试单元** | | | StockGradeMapper | |
| **是否有测试代码** | | 是 | **测试代码** | StockGradeMapperTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getStockGrade(String id)方法，并打印返回的结果 | sh600000对应评分 | sh600000对应评分 | Pass |
| 1 | 以”sh600000**”**为参数，调用getCurrentInfo(String id)方法，并打印返回的结果 | sh600000对应评分，以及对应的原始数据 | sh600000对应评分，没有返回对应的原始数据 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-008 | |
| **测试单元** | | | StockInfoMapper | |
| **是否有测试代码** | | 是 | **测试代码** | StockInfoMapperTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getStockInfo(String id)方法，打印返回结果 | sh600000对应公司数据 | sh600000对应公司数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-29 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-011 | |
| **测试单元** | | | FavouriteStockImpl | |
| **是否有测试代码** | | 是 | **测试代码** | StockFavouriteTest |
| **前提条件** | 输入收藏股票列表 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**,“sh600004”组成的list为参数为参数，调用getFavourite(ArrayList)方法，并打印返回的结果 | sh600000,sh6000004,对应五天的数据 | sh600000,sh6000004,对应五天的数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-012 | |
| **测试单元** | | | ForecastDataImpl | |
| **是否有测试代码** | | 是 | **测试代码** | ForecastTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getForecastData(String id）方法并打印结果 | sh600000下一天的预测数据 | sh600000下一天的预测数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-013 | |
| **测试单元** | | | GradeImpl | |
| **是否有测试代码** | | 是 | **测试代码** | GradeTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getCurrentInfo(String id）方法并打印结果 | sh600000各项评分数据 | sh600000近十个交易日评分原始数据 | Pass |
| 2 | 以”sh600000**”**为参数，调用getCurrentInfo(String id）方法并打印结果，输出VO内原始数据 | sh600000近十个交易日评分原始数据 | sh600000近十个交易日评分原始数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-014 | |
| **测试单元** | | | IndustryImpl | |
| **是否有测试代码** | | 是 | **测试代码** | IndustryImplTest |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 调用getIndustryAnalysis()方法并打印结果 | 所有行业的评分数据 | 所有行业的评分数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-015 | |
| **测试单元** | | | RelativeImpl | |
| **是否有测试代码** | | 是 | **测试代码** | RelativeImplTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh600000”调用getRelativeData(String id)方法，并打印结果 | sh600000的大盘相关性 | sh600000的大盘相关性 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-016 | |
| **测试单元** | | | SingleInfoImpl | |
| **是否有测试代码** | | 是 | **测试代码** | SingleInfoTest |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | getSingleInfo()方法，并打印结果 | 所有股票的简要信息 | 所有股票的简要信息 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-016 | |
| **测试单元** | | | StockComparisionImpl | |
| **是否有测试代码** | | 是 | **测试代码** | SingleInfoTest |
| **前提条件** | 无 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以“sh600000”调用getStockComparsion(String id)方法，并打印结果 | sh600000股票对比所需信息 | spring异常 | Fail |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-017 | |
| **测试单元** | | | StockInfoImpl | |
| **是否有测试代码** | | 是 | **测试代码** | StockInfoTest |
| **前提条件** | 输入股票信息 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getStockInfo(String id)方法，打印返回结果 | sh600000对应公司数据 | sh600000对应公司数据 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-018 | |
| **测试单元** | | | TabTableDataImpl | |
| **是否有测试代码** | | 是 | **测试代码** | TabTableDataImplServiceTest |
| **前提条件** | 输入股票信息、日期 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getTablesInfo(String id , LocalDate start , LocalDate end)方法，打印返回结果 | sh600000在调用时间内的详细信息 | sh600000在调用时间内的详细信息 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | | | JunitAnyQuant3-019 | |
| **测试单元** | | | TabTableInstructionImpl | |
| **是否有测试代码** | | 是 | **测试代码** | TabTableInsTest |
| **前提条件** | 输入股票信息、日期 | | | |
| **序号** | **步骤及操作** | **预期结果** | **实际结果** | **Pass/Fail** |
| 1 | 以”sh600000**”**为参数，调用getTablesInfo(String id , LocalDate start , LocalDate end)方法，打印返回结果 | sh600000在调用时间内的详细信息，以及tabtable分析信息 | sh600000在调用时间内的详细信息，以及tabtable分析信息 | Pass |
| **测试人员** | 张晨剑 | **测试日期** | 2016-5-30 | |
| **测试用例实际执行成本** | | 0.1人日 | | |

迭代三单元测试结论

单元测试第一次通过率为77.27%，发现5个问题，问题主要存在调用sql语句的column名出错、spring框架运用等方面。

经调试后，测试用例全部通过，可以正常运行。

迭代三集成测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 1 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 输入股票信息正确，网络连接正常 | | | | |
| 步骤 | 1.添加股票进入对比页面  2.进入股票对比页面 | | | | |
| 预期结果 | 显示股票各项数据的对比 | | | | |
| 实际结果 | 显示股票各项数据的对比 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 2 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 保存用户收藏记录，网络连接正常 | | | | |
| 步骤 | 1.进入股票收藏页面 | | | | |
| 预期结果 | 显示大盘和收藏股票的简单信息 | | | | |
| 实际结果 | 显示大盘和收藏股票的简单信息 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 3 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 网络连接正常 | | | | |
| 步骤 | 1.进入行业评分页面  2.点击行业 | | | | |
| 预期结果 | 显示行业评分数据、排名及行业内股票评分、排名 | | | | |
| 实际结果 | 显示行业评分数据、排名及行业内股票评分、排名 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 4 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 个股信息输入正确，网络连接正常 | | | | |
| 步骤 | 1.进入个股分析页面  2.找到数据分析项 | | | | |
| 预期结果 | 显示个股实时数据分析 | | | | |
| 实际结果 | 显示个股实时数据分析 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 5 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 个股信息输入正确，网络连接正常 | | | | |
| 步骤 | 1.进入个股信息页面  2.查看各项图标 | | | | |
| 预期结果 | 显示个股各项图表 | | | | |
| 实际结果 | 显示个股各项图表 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 6 | 涉及模块 | controller,dao,impl,mapping,pojo,service,vo | | |
| 依赖条件 | 个股信息输入正确，网络连接正常 | | | | |
| 步骤 | 1.进入个股信息页面  2.查看实时数据，分时图及表格 | | | | |
| 预期结果 | 显示个股实时数据 | | | | |
| 实际结果 | 显示个股实时数据 | | | | |
| Pass/Fail | pass | 测试者 | 张晨剑 | 执行成本 | 0.1人日 |

迭代三集成测试结论

集成测试通过率为100%，经分析，由于单元测试的存在减少了编码错误，提高了集成测试的通过率。

界面层初步完成，需要经过大量的优化。优化后，原先的集成测试同样可以通过，保持高通过率。