

Finsight 金融投资组合管理系统

项目交付文档

版本: 1.0.0

交付日期: 2025-07-29

代码仓库: <https://github.com/hxtOvO/Finsight>

团队: Finsight开发组

目录

1. 项目概述
2. 技术架构
3. 核心功能说明
 - 后端API功能
 - 前端交互功能
4. 环境部署指南
5. 使用手册
6. 接口文档
7. 项目维护与扩展
8. 团队协作与版本控制

1. 项目概述

1.1 项目背景

随着个人金融资产多样化，用户对“股票、债券、现金等资产的集中管理”需求日益凸显。本项目旨在开发一套轻量化金融投资组合管理系统，帮助用户实时监控资产状态、分析投资绩效，并简化资产配置操作。

1.2 项目目标

- 构建稳定可靠的投资组合管理REST API，支持资产数据的存储、查询与更新。
- 开发直观易用的前端交互界面，实现资产可视化与操作便捷性。
- 满足核心业务场景：资产状态浏览、绩效趋势分析、资产增减管理。

1.3 核心价值

- 数据聚合**: 统一管理多类型资产，避免分散记录的繁琐。
- 实时洞察**: 通过可视化图表直观展示资产历史趋势与当前状态。
- 轻量化设计**: 无需复杂配置，快速部署并投入使用。

2. 技术架构

2.1 整体架构

采用“前后端分离”架构，通过REST API实现数据交互，架构图如下：



2.2 技术栈详情

模块	技术选型	核心作用
后端	Node.js (Express)	构建REST API, 处理业务逻辑
数据库	MySQL 8.0	存储资产数据、历史绩效、股票信息
前端	H5 + JavaScript + Chart.js	页面渲染与数据可视化
数据交互	Axios	前端请求后端API、后端调用第三方数据
接口文档	Swagger (swagger-jsdoc)	自动生成API文档, 支持在线调试
环境配置	dotenv	管理环境变量 (数据库账号、端口等)
版本控制	Git + GitHub	代码管理与团队协作

2.3 数据模型设计

核心数据模型围绕“资产”“绩效”“股票”三大实体设计，关键表结构如下：

表名	核心字段	作用
current_assets	type(资产类型)、symbol(股票代码)、amount(数量)	存储当前持有的资产信息
portfolio	total_value(总资产)、gain_loss(盈亏)	存储投资组合核心指标
performance_history	date(日期)、value(价值)、range_type(时间范围)	存储资产绩效历史数据
featured_stocks	symbol(股票代码)、price(价格)、change_percent(涨幅)	存储股票实时数据

3. 核心功能说明

3.1 后端API功能

3.1.1 资产数据管理

接口	功能描述	核心逻辑
GET /api/assets	获取当前资产分布 (现金/股票/债券/其他)	聚合current_assets数据, 计算股票实时价值
PUT /api/assets/:type	更新指定类型资产价值 (如增加现金、调整股票数量)	校验资产合法性 (非负), 更新后同步总资产

接口	功能描述	核心逻辑
GET /api/portfolio	获取投资组合核心指标（总资产、盈亏）	实时计算总资产，对比历史基准计算盈亏

3.1.2 股票数据管理

接口	功能描述	核心逻辑
GET /api/featured-stocks	获取自选股票列表（含价格、涨跌幅）	查询Featured_stocks表，返回格式化数据
POST /api/featured-stocks/add	添加股票到自选列表（自动获取实时价格）	调用第三方API获取价格，写入数据库
POST /api/featured-stocks/remove	从自选列表删除股票	根据股票代码删除Featured_stocks记录

3.1.3 绩效分析

接口	功能描述	核心逻辑
GET /api/performance/:range	获取资产总绩效趋势（支持7天/1月/6月）	从performance_history筛选时间范围数据
GET /api/assets/:type/performance/:range	获取单类型资产绩效（如股票单独趋势）	按资产类型筛选asset_history数据

3.2 前端交互功能

3.2.1 资产可视化

- 资产分布仪表盘**: 通过环形图展示现金、股票、债券、其他资产的占比。
- 绩效趋势图表**: 使用折线图展示指定时间范围（7天/1月/6月）的资产价值变化。
- 实时数据卡片**: 显示总资产、盈亏金额、盈亏百分比等核心指标。

3.2.2 资产操作

- 资产调整**: 支持现金增减、股票数量调整（买入/卖出）。
- 股票管理**: 添加自选股票、删除无需监控的股票。
- 数据筛选**: 通过时间范围切换（7天/1月/6月）查看不同周期的绩效数据。

3.2.3 用户体验优化

- 响应式设计**: 适配不同设备屏幕（PC/平板/手机）。
- 交互反馈**: 操作后实时更新数据并显示成功提示。
- 隐私模式**: 支持隐藏敏感金额（如将“10000元”显示为“***”）。

4. 环境部署指南

4.1 前置依赖

依赖项	版本要求	安装说明
Node.js	v16.0.0+	推荐使用nvm管理版本, 官网下载
MySQL	8.0+	需启动服务 (Windows: <code>net start mysql80</code>)
npm	v8.0.0+	Node.js自带, 无需单独安装
Git	任意稳定版本	用于拉取代码仓库

4.2 部署步骤

步骤1：拉取代码

```
# 克隆仓库
git clone https://github.com/hxt0v0/Finsight.git
cd Finsight
```

步骤2：配置环境变量

1. 在项目根目录创建.`env`文件, 模板如下:

```
# 数据库配置
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=your_mysql_password
DB_NAME=finsight
DB_PORT=3306

# 服务配置
PORT=3000
```

步骤3：安装依赖并启动服务

```
# 安装后端依赖
cd backend
npm install

# 启动后端服务 (自动初始化数据库)
npm start
# 预期输出: "🚀 FinSight Backend running on http://localhost:3000"

# (新终端) 启动前端 (无需额外安装, 直接通过后端静态文件服务访问)
# 前端页面地址: http://localhost:3000
```

步骤4：验证部署成功

- 访问后端服务：<http://localhost:3000/api/health>，返回{"status": "OK"}。
- 访问前端页面：<http://localhost:3000>，显示资产仪表盘。
- 访问接口文档：<http://localhost:3000/api-docs>，显示Swagger界面。

5. 使用手册

5.1 核心功能使用流程

流程1：查看资产状态

1. 打开前端页面，默认显示“资产分布仪表盘”。
2. 查看环形图：了解现金、股票、债券、其他资产的占比。
3. 查看核心指标：页面顶部显示总资产、盈亏金额及百分比。

流程2：管理自选股票

1. 进入“股票管理”模块，查看当前自选股票列表。
2. 添加股票：点击“添加股票”，输入股票代码（如AAPL），点击确认。
3. 删除股票：在股票列表中，点击目标股票后的“删除”按钮。

流程3：分析绩效趋势

1. 在“绩效分析”模块，默认显示最近7天的资产趋势。
2. 切换时间范围：点击“1月”或“6月”按钮，查看对应周期的趋势图。
3. 查看单资产绩效：点击资产类型（如“股票”），单独显示该类型资产的趋势。

流程4：调整资产配置

1. 进入“资产调整”模块，选择资产类型（如“现金”）。
2. 输入调整金额（正数为增加，负数为减少），点击“确认调整”。
3. 系统自动更新总资产及相关图表数据。

5.2 常见操作示例

示例：买入股票并查看效果

1. 在“股票管理”添加“NVDA”（英伟达）股票。
2. 在“资产调整”选择“stock”类型，输入股票代码“NVDA”和数量“5”（表示买入5股）。
3. 查看“资产分布”：股票资产占比增加；查看“绩效图表”：总资产实时更新。

6. 接口文档

系统集成Swagger接口文档，支持在线调试。

- **访问地址：**<http://localhost:3000/api-docs>（启动服务后可访问）
- **文档内容：**包含所有API的请求参数、响应格式、错误码说明。
- **使用方式：**在文档页面选择接口，点击“Try it out”输入参数，点击“Execute”发送请求。

7. 项目维护与扩展

7.1 日常维护

- **数据备份**: 定期备份MySQL数据库 (推荐每日凌晨执行) :

```
# 备份命令示例
mysqldump -u root -p finsight > finsight_backup_$(date +%Y%m%d).sql
```

- **日志查看**: 后端运行日志可通过console输出查看, 关键错误会标记“X”前缀。
- **第三方API监控**: 若股票数据获取失败, 检查RAPIDAPI_KEY有效性 (需定期更新)。

7.2 功能扩展建议

潜在需求	扩展方案
多用户支持	添加用户表 (users), 通过用户ID隔离数据
资产交易记录	新增transactions表, 记录买入/卖出明细
更复杂的绩效分析	集成均线计算、年化收益等指标
前端框架升级	迁移至Vue/React, 提升组件复用性

8. 团队协作与版本控制

8.1 分支管理策略

- **main**: 主分支, 仅合并经过测试的稳定代码。
- **dev**: 开发分支, 团队成员从该分支创建功能分支。
- **feature/xxx**: 功能分支 (如feature/stock-add), 完成后合并到**dev**。

8.2 代码提交规范

提交信息格式: [类型] 描述, 示例:

- [feat] 添加股票绩效分析接口
- [fix] 修复资产调整负数校验bug
- [docs] 更新部署文档的环境变量说明

8.3 协作工具

- **代码管理**: GitHub (仓库权限: 团队成员均为开发者权限)
- **任务跟踪**: 推荐使用Trello, 分“待办”“进行中”“已完成”列表管理任务。

附录: 常见问题解决

1. **数据库连接失败**: 检查.env配置是否正确, MySQL服务是否启动。
2. **股票数据无法获取**: 检查RAPIDAPI_KEY是否有效, 网络是否通畅。
3. **前端页面空白**: 确认后端服务已启动, 访问<http://localhost:3000>而非前端单独地址。

文档责任人：Finsight开发组

联系方式：通过GitHub仓库Issue反馈问题