DolphinDB 白皮书

Starfish 因子开发管理平台

DATABASE
ANALYTICS
STREAMING

内容

第 1 章. Starfish 因子开发管理平台

Starfish 因子开发管理平台，基于 DolphinDB 强大的数据库和计算引擎，专为金融机构投研场景量身打造。平台汇聚因子开发、策略验证、绩效分析于一体，旨在成为量化分析师和投研团队的全方位助手。核心功能包括：

- **因子计算**：支持各类因子模型的计算，帮助用户快速构建投资因子。
- **评价与分析**：提供因子的评估工具，确保用户可以清晰地判断因子的有效性。
- **策略回测**：灵活的回测引擎，支持多品种市场数据，帮助用户进行真实市场环境中的策略验证。
- **绩效归因**：支持多维度的业绩归因分析，帮助用户深度剖析收益来源与风险暴露。
- **工作流支持**：完整的工作流管理工具，简化从因子研究、策略开发到执行的每一步，提升效率。

Starfish 因子开发管理平台为用户提供从因子构建、策略验证到绩效追踪的全程支持，赋能量化分析师迅速抓住市场机遇，优化因子策略。依托 DolphinDB 的高效计算引擎和简洁直观的工作流设计，平台极大提升了研究效率，帮助用户在复杂的市场环境中作出更为精准的投资决策。

第 2 章. 系统架构

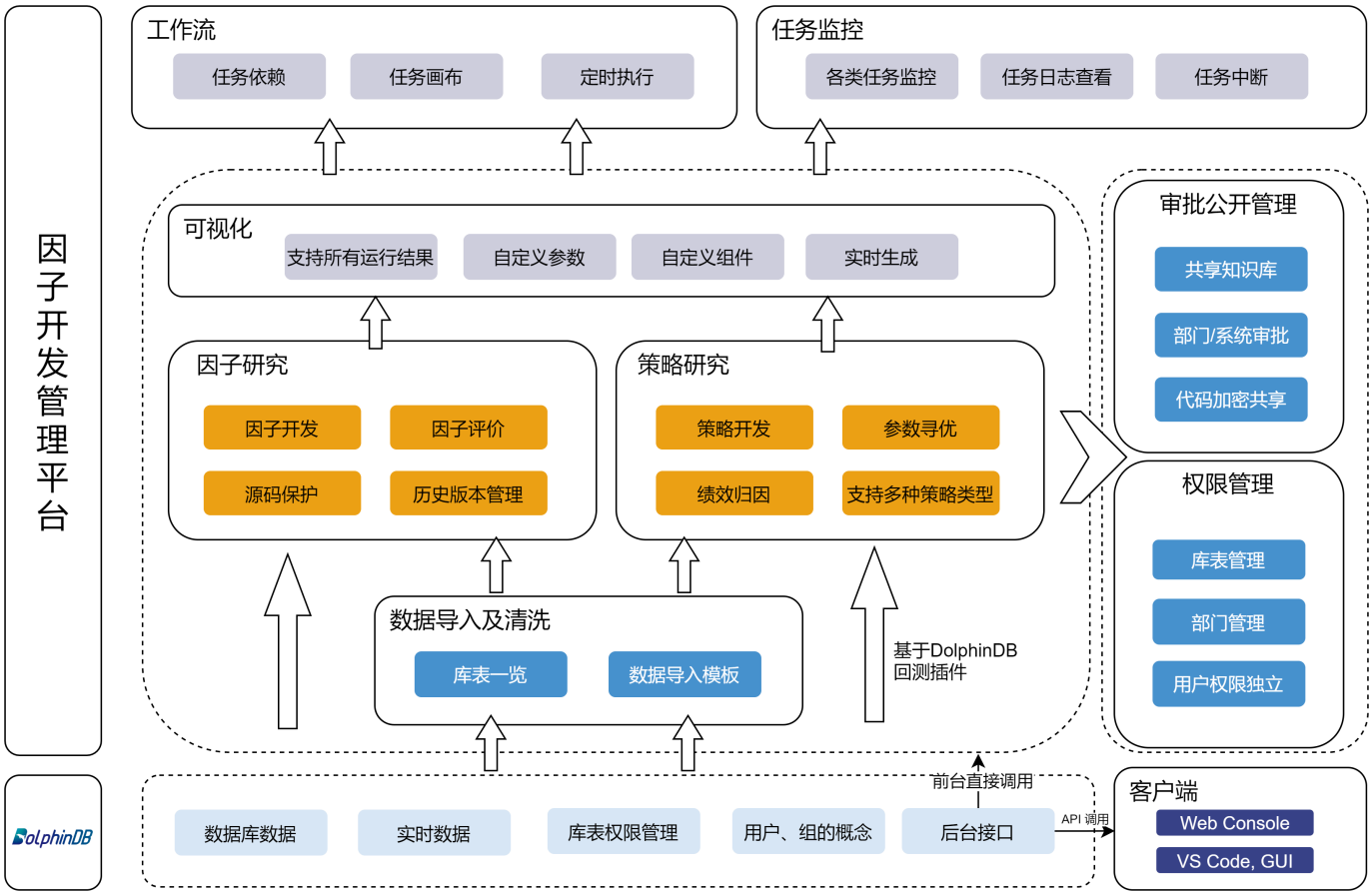

图2-1 Starfish 系统架构图

2.1 设计框架

2.1.1 前后端架构

Starfish 因子开发管理平台的架构分为前端和后端模块。后端以 DolphinDB 强大的计算引擎为核心，提供高效的数据处理与计算服务；前端则通过 JavaScript API 直接调用后端接口，以便实现低延迟的数据交互和实时响应。

后端架构与 DolphinScript API

后端采用 Dolphin Script 编写独立的函数接口，通过模块化设计提升了代码管理的简洁性和系统的扩展性。接口遵循统一的 API 规范，确保功能模块间的兼容性和调用的便捷性，也能够快速响应业务需求的变化，实现高效的功能迭代。后端的接口最终是通过模块以及函数视图提供给用户部署使用的。这样的后端架构有以下的优势：

- **DolphinScript API**：后端的 API 接口是通过 DolphinDB 独有的 DolphinScript 编写的，形成模块和函数视图。DolpphinScript 可以直接访问 DolphinDB 数据库和计算引擎，进行因子计算、策略回测等核心任务。

- **性能和并发**：基于 DolphinDB 高效的存储和分布式计算框架，DolphinScript能够快速处理大规模数据，同时支持高并发调用。所有任务均采用后台任务的方式运行，不会堵塞前台进程调用。

前端架构与 JavaScript API

前端采用 TypeScript 编写，结合 React 和 Ant Design 组件库，通过组件化设计和层次分明的架构，实现了灵活的界面管理，并高效响应用户需求。TypeScript 为前端提供了类型安全和更优的开发体验，与后端通过 JavaScript API 进行连接和数据交互，这样的前端架构具有以下优势：

- **TypeScript 代码结构与安全性**：TypeScript 的类型检查确保了代码的可靠性，使开发团队能够在编写时发现潜在错误，并提供更好的代码维护性，特别是在大型应用中提升了稳定性和可读性。

- **JavaScript API 数据交互**：前端通过 JavaScript API 连接 DolphinDB 并调用 DolphinScript 后端接口，发起数据请求并接收计算结果，从而实现前端与后端的无缝对接。

- **响应式设计**：借助 JavaScript API，前端实现低延迟的响应和即时数据更新，为用户带来流畅的操作体验。

- **用户交互与数据展示**：前端提供丰富的图表、表格等展示工具，使复杂的因子分析、回测分析结果更加直观，便于用户理解和操作。

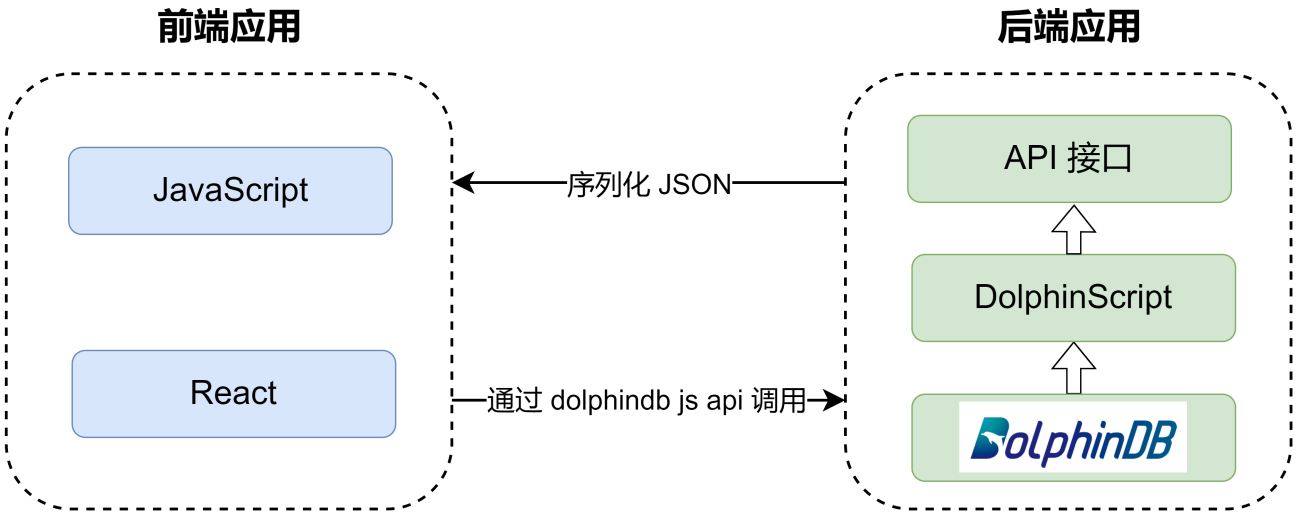

图2-2 Starfish 前后端

综上所述，Starfish 因子开发管理平台采用前后端分离架构。后端使用 DolphinDB 数据库与 DolphinScript 构建模块化函数接口，支持因子计算、策略回测等核心任务，具有高性能并发处理能力。前端使用 TypeScript 编写，结合 React 和 Ant Design 提供用户界面，并通过 JavaScript API 实现数据交互。前后端的架构保证了系统的灵活性和高效响应，使用户操作体验流畅、清晰。

2.1.2 权限管理

2.1.2.1 用户、组、角色和权限

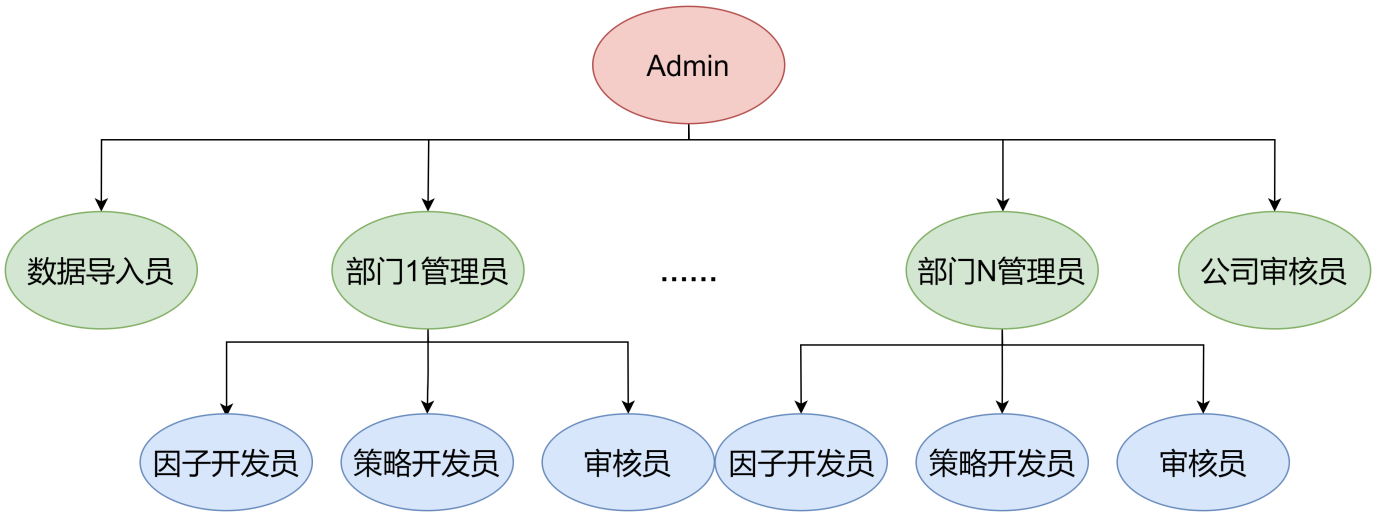

图2-3 Starfish 的用户、组和角色

在因子平台中，有用户和部门两个概念：

- 因子平台中的用户即为 DolphinDB 数据库中的用户，在因子平台中新建用户” user1”，会在数据库中也新建” user1” 这个用户；
- 因子平台中的部门即为 DolphinDB 数据库中的组，在因子平台中新增部门ID为” dept1 “的部门，会在数据库中新建名为” facplf_dept1” 的组。

在因子平台中，用户的等级可以分为三种：

- admin：系统管理员，负责管理系统中所有的用户和部门，其权限包含：创建新用户，创建或分配用户的组别，所有权限的管理，删除或禁用激活用户；是 DolphinDB 数据库中的admin。
- 部门管理员：负责管理部门下所有用户的因子、策略开发及审核权限；是 DolphinDB 数据库中的普通用户。
- 普通用户：没有管理权限，但可以根据管理员分配的权限使用因子平台；是 DolphinDB 数据库中的普通用户。

激活、禁用、创建及删除用户与组的功能：

- 因子平台的用户不会自动将数据库的用户纳入平台的用户中去，需要手动在因子平台的用户管理页面中激活这些用户。如下图2-4所示，高亮用户为平台用户，名称为灰色的用户为非激活的数据库用户 。管理员点击启用按钮，即可激活用户对于平台的使用权；管理员也可以对已激活的用户进行禁用因子平台使用权的操作。
- 管理员可以在管理页面进行创建新用户及删除已有用户的操作，这些操作不仅会删除或创建因子平台的用户、组，也会删除或创建 DolphinDB 数据库中的用户、组。

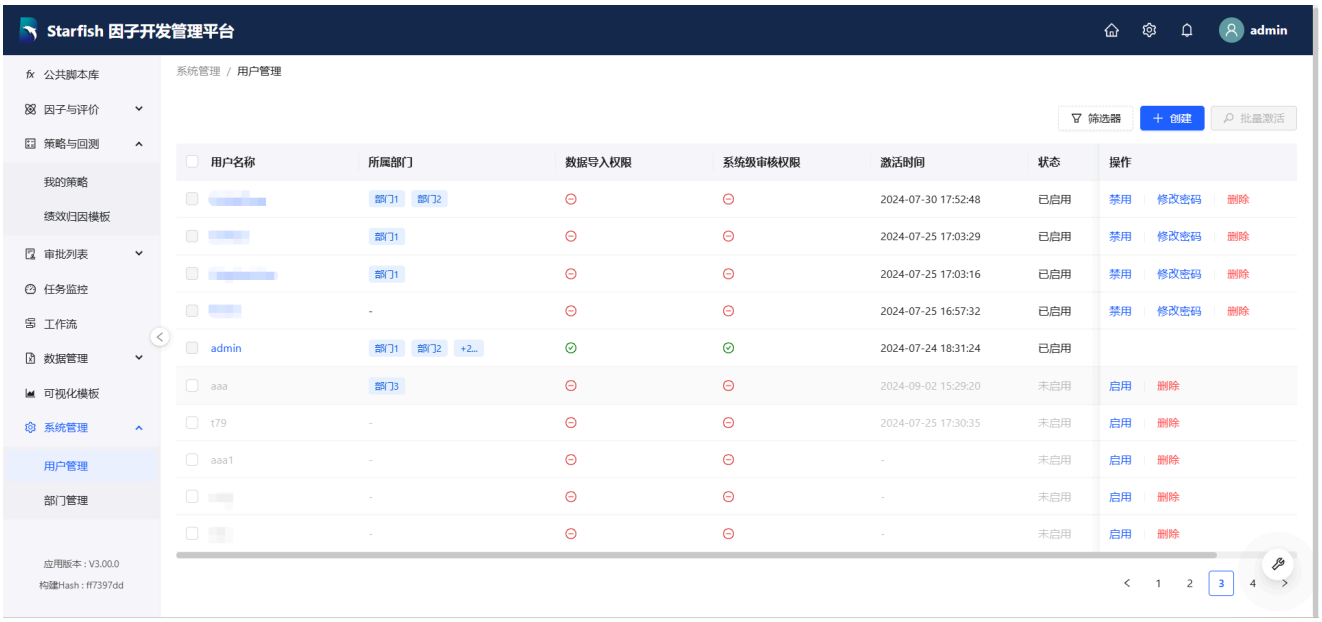

图2-4 Starfish 因子平台的用户管理页面

在因子平台中，用户的角色分为系统角色和部门角色两大类：

- 系统角色包括审核权限：
  - 审核权限允许用户作为系统审核员，对提交到系统的公开申请进行审核。
- 部门角色包括因子模块、策略模块和审核角色：
  - 因子模块角色允许用户使用因子库版块功能。
  - 策略模块角色允许用户使用策略回测版块功能。
  - 审核权限角色用户作为该部门审核员，对提交到该部门的公开申请进行审核。

在因子平台中，每个用户的数据库权限与在DolphinDB 环境中的权限相同。在因子平台数据管理页下可以实现用户对于数据库、表的权限查看和管理：

- 部门管理员和拥有数据导入权限的用户可以对自己管理的库表进行权限的管理和监控。
- admin 可以管理和设置用户和部门能够创建的数据库的前缀，为数据的有序管理提供支持。

总结来说，因子平台的权限管理是基于 DolphinDB 已有的用户权限进行的上层实现。因子平台中的用户和部门即为 DolphinDB 中的用户和组；因子平台用户对于数据库表的权限即为用户对 DolphinDB 环境中的数据库表权限。而因子、策略开发、审核权限为因子平台中特有的权限类别，平台中的部门管理员也是因子平台中独有的用户角色。

2.1.2.2 代码版本及加密

代码版本管理

在因子平台中，通常每个用户拥有自己独立的工作空间，因子平台集成了完整的代码版本控制系统，支持策略和因子代码的历史版本查看和管理。用户可以轻松回溯和审查历史代码版本，确保策略的稳定性和可追溯性。

代码加密

平台中所有的代码都会以二进制加密的方式保存在库内，因此就算是 admin 也无法直接通过读表的方式看到真实代码。

除此之外，平台提供灵活的代码加密选项，用户可以选择在代码提交、审核和公开过程中是否对代码进行加密。如果选择加密，即使代码被公开，也无法被未授权的用户查看源代码内容，但仍可以被授权的用户执行。这一功能可以保护投资策略的知识产权同时促进策略的共享和复用。

2.1.3 关键技术特性

全面的模板化功能：因子平台通过提供多种编辑模板，包括因子计算模板、评价模板、策略的绩效归因模板、可视化模板和数据导入模板，显著简化了用户的操作流程并降低了开发与测试的复杂性。这些模板设计得既灵活又功能丰富，使用户仅需通过简单的参数设定，便能迅速启动并运行预定的模板，从而实现高效且准确的因子分析和策略回测。

严格的角色和权限管理：因子平台设计了清晰的角色分工，涵盖从系统管理员到部门管理员，再到普通用户。每个级别的用户都赋予了明确的职责和权限，形成了严密的权限管理体系。系统管理员拥有最高权限，负责整个平台的用户和权限管理，确保系统的安全性和效率。部门管理员主要负责监督部门内的策略和因子开发，以及审核流程，确保部门内活动的顺畅运行。普通用户则主要聚焦于日常的因子分析和策略操作，依据管理员分配的权限进行各项数据活动。

灵活的 API 支持：平台内实现了一系列 API 来支持从数据导入到因子计算、策略回测、结果可视化等全方位的量化分析。用户可以通过各类语言，如python、java等，使用 API 进行自定义的开发，大幅缩短开发实现的周期。

高效的脚本管理：平台允许用户在因子平台上方便地创建、编辑、试运行和部署各种脚本。支持版本控制，确保脚本的稳定性和可维护性。同时，定时执行和任务监控功能进一步提高了操作的便捷性和脚本的实用性。

丰富的可视化分析工具：因子平台集成了多种的数据可视化组件，提供了丰富的图表和报表选项，帮助用户直观地分析数据和评估因子或者策略的表现。可视化工具支持自定义配置，使用户能够根据需要调整结果格式，更好地洞察数据背后的趋势和模式。

2.2 部署方式

Starfish 因子开发管理平台的包内有以下内容：

```
starfish
├──  modules
|    └──  starfish
|        |──  facplfBasic.dom
|        ├──  facplf.dom
|        ├──  ...
├──  web
|    └──  starfish
|        |──  vs
|        ├──  assets
|        ├──  ...
```

```
├──  starfish
|    └──  init
|        |──  init.dos
|        ├──  ...
```

因子平台可以部署在 DolphinDB 单节点模式或集群模式中。以下是这两种模式的具体部署步骤：

注： 因子平台需使用专用 license，请联系技术支持。

2.2.1 单节点模式部署

- 解压文件：
  - 使用命令 `unzip starfish.zip -d /home/dolphindb/server/` 将文件解压到指定目录。
- 安装必备插件并增加配置项：
  - 从 Dolphindb 官方插件市场网站下载并安装 Backtest 和 MatchingEngineSimulator 插件。
  - 在 dolphindb.cfg 中增加 `preloadModules=plugins::Backtest,plugins::MatchingEngineSimulator`
- 初始化：
  - 连接至单节点并执行`run("/home/dolphindb/server/starfish/init/init.dos")`

2.2.2 集群模式部署

- 解压文件：
  - 与单节点模式相同，将文件解压到 /home/dolphindb/server/。
- 文件分发：
  - 将 web、modules 和 starfish目录分别放置到每个数据节点和计算节点的对应目录。
- 安装必备插件并增加配置项：
  - 在所有控制节点、数据节点和计算节点上安装 Backtest 和 MatchingEngineSimulator。
  - 在 controller.cfg，cluster.cfg 中增加 `preloadModules=plugins::Backtest,plugins::MatchingEngineSimulator`。
- 初始化：
  - 只需在任一数据节点或计算节点执行一次初始化脚本。

2.3 使用方式

2.3.1 前台调用

部署完成后，可通过 http://ip:port/starfish/index.html 访问因子平台前端应用。 其中 ip 是节点所在服务器的 IP，port 是节点的端口号，使用时请根据实际情况调整。

对于集群模式部署的因子平台，可通过任一数据节点或计算节点访问。

进入因子平台前台应用后，用户即可以通过简单的点击和拖拽操作来管理因子、执行策略回测、查看分析结果等。这种方式特别适合对 DolphinDB 脚本不太熟悉的用户，可以直观地操作和获取所需信息，实现因子分析和策略回测结果。例如，首页中即包含了当前用户的因子和策略数量以及运行任务等信息：

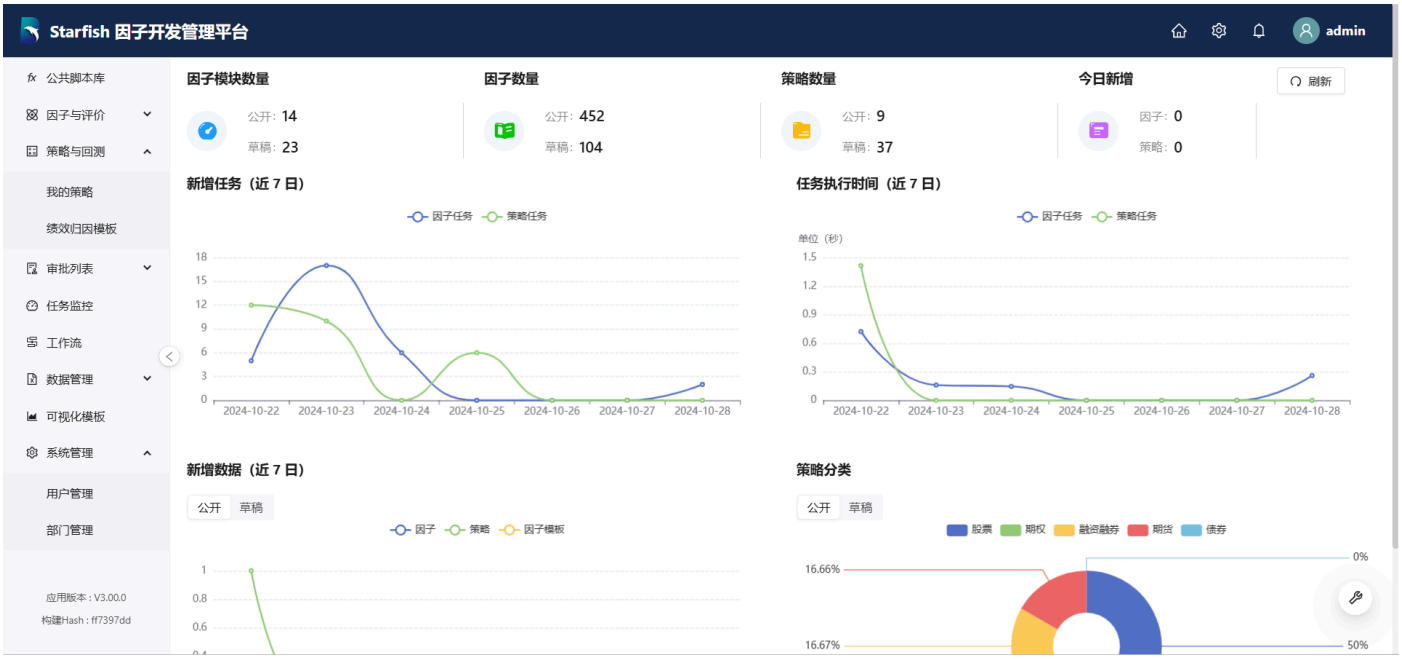

图2-5 Starfish 因子平台首页

2.3.2 API 调用

用户除了借助 web 应用使用 Starfish 因子开发管理平台之外，也可以调用 API 接口与平台做交互，支持各类语言调用，适用于需要高度自定义的场景。开发者可以通过 API 执行所有前台可以完成的操作，例如因子上传及测试、策略开发及回测、结果查询等。例如，用户在 python 端需要查看自己的草稿因子，可以用以下方法获得结果：

```
import dolphindb as ddb
s = ddb.session()
s.connect(host="localhost", port=8848, userid="admin", password="123456")
strategy_list = s.run("starfish::facplfBasic::facplf_get_factor_list()")
```

返回的结果为一个字典，包含 items 和 total 两个字段，分别代表策略列表的表格和策略总数，因此我们可以使用以下的代码在 python 中获取到当前登录用户的所有策略列表：

```
strategy_list["items"]
```

	id	strategy_name	strategy_group	data_type	msg_as_table	create_time	update_time	comment	tags	creator	run_record_count
0	9e6781be-2921-0c9b-c35c-6b9d7d048192	onBarCTAdemo	stock	3	False	2024-08-14 10:36:45.561	2024-10-31 14:26:28.529			admin	64
1	a3df58db-ab95-9f8c-e08b-3da51813457f	API_demo1	stock	1	False	2024-10-29 11:10:09.843	2024-10-31 13:42:30.680			admin	0
2	45f4ee13-4555-4ce1-3c3f-4039c04d8f73	APIDemo	stock	1	False	2024-10-25 17:38:45.102	2024-10-25 17:39:24.062			admin	0
3	1a5d079b-eb10-9b7c-42e1-d356c1009fdf	API_demo_real_time	futures	3	False	2024-10-23 17:09:00.659	2024-10-23 17:09:00.716			admin	3
4	39cd8c97-4a2d-13ab-da23-4f4450e6b4aa	API_demo	stock	1	False	2024-10-23 13:50:51.896	2024-10-23 15:52:04.309			admin	5

图2-6 facplf_get_factor_list调用结果

第 3 章. 产品功能

本产品面向基金、证券、期货等机构投资者，基于 DolphinDB 强大的库内开发能力以及优异的性能表现，提供了一套高效的因子及策略开发平台。平台中集成了因子计算、因子评价、回测、组合优化等功能，使研究员可以高效完成从因子开发到投资组合应用的全流程工作。

3.1 公共脚本库

公共脚本库是因子平台的核心功能模块之一，旨在集中管理和访问所有可公开使用的因子、策略及函数模块。这一功能的设计促进了资源共享，显著提高了开发效率，特别适用于因子和策略的开发场景。用户能够高效地进行搜索、测试和应用这些资源，从而加快投资决策过程，同时避免了重复开发工作。

- **通用功能**
  - **资源管理：**提供一个集中式的界面，方便用户查找和管理可公开的因子和策略。
  - **搜索与筛选：**内置高效的搜索引擎，帮助用户快速找到符合需求的脚本。
- **因子模块**
  - **内置因子库**：默认集成了七种 DolphinDB 提供的、在金融市场分析中被广泛应用的系统级因子模块（mytt、alpha191等）。
  - **批量测试**：用户可以在此页面进行因子的批量测试，例如，可以评估某个alpha因子在特定股票池中的表现，以评估其在实际投资决策中的有效性。

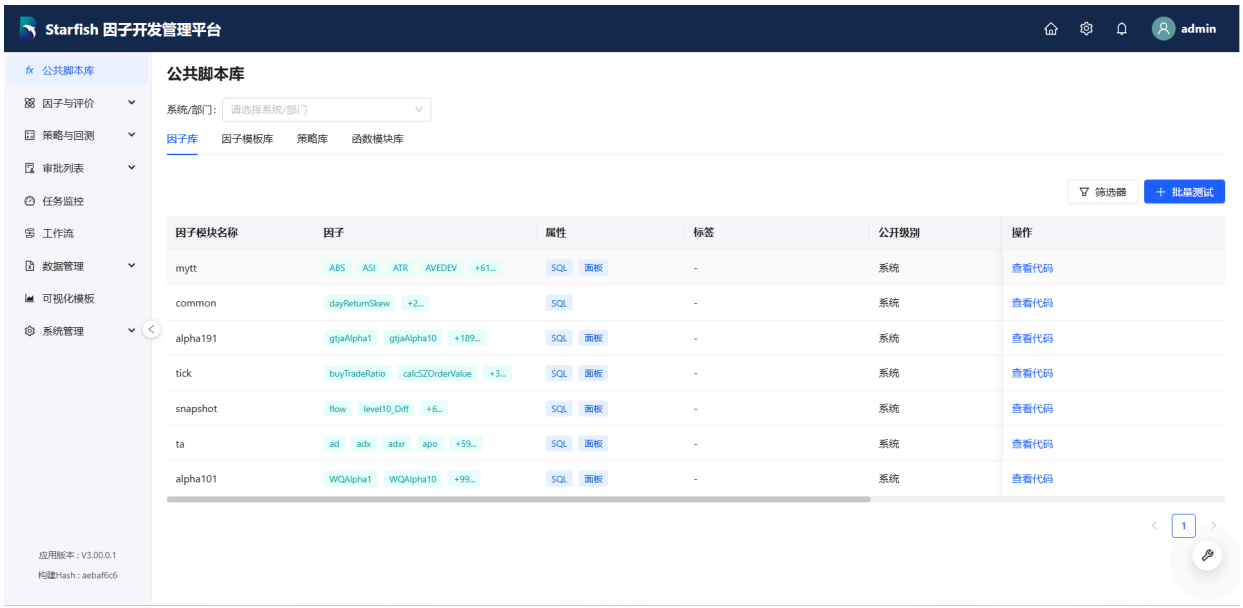

图3-1 公共脚本库中的内置因子模块

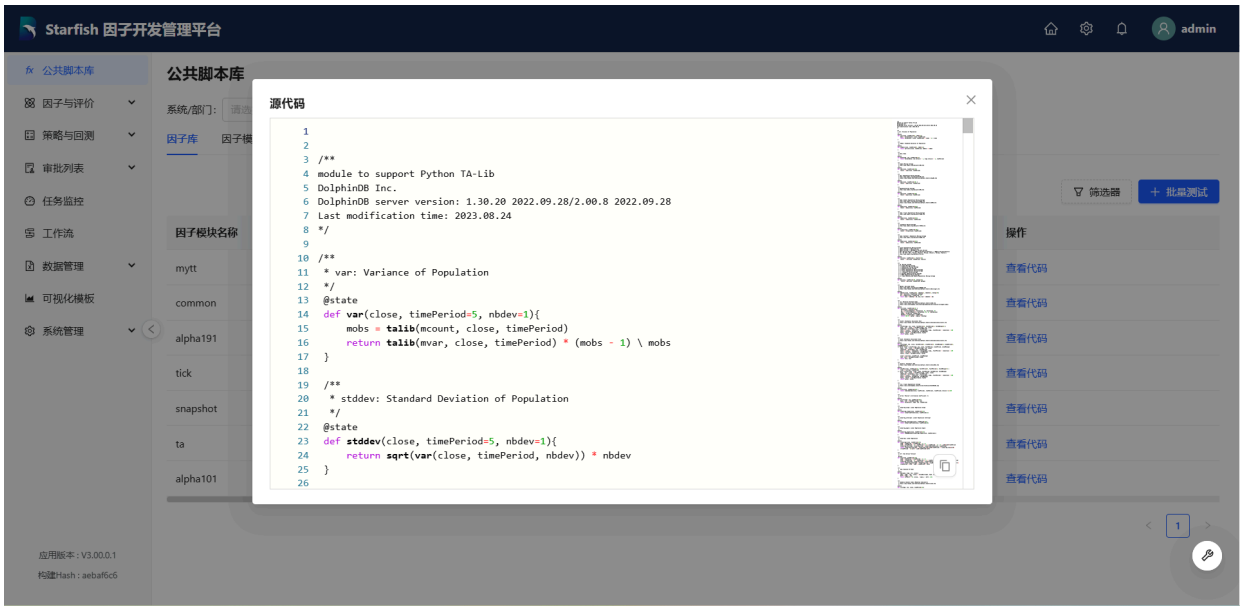

图3-2 内置 ta 因子源代码

- **策略库**
  - **公开策略库**：为用户提供一个展示和测试公开回测策略的平台，用户可以查看不同策略的具体实现和历史回测结果。
  - **定制与优化**：允许用户选择与自己投资风格相符的策略进行定制和优化，并在自己的资产组合上进行测试。
- **函数模块库**

- **函数复用**：存放了平台所有用户创建的各种公共函数模块，其他用户可以直接引用这些函数，避免重复编写代码，加快新因子或策略的开发周期。
- **内置函数**：包括 alphalens 因子评价函数、kDay 模拟数据生成函数、通联数据导入函数、Brinson 和 Campisi 的计算函数，用户可以根据需求在这些函数基础上进行开发。

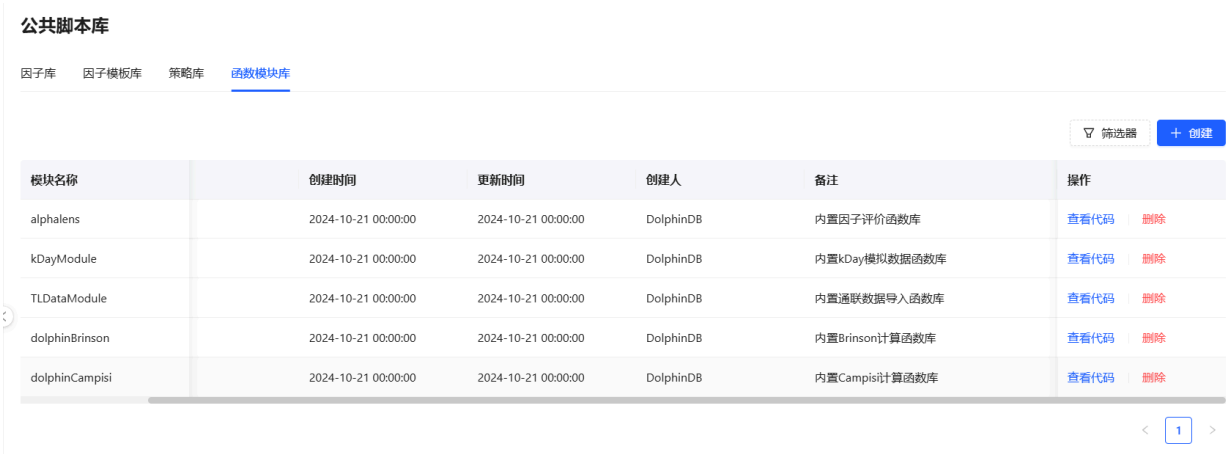

图3-3 公共脚本库中的内置函数库

3.2 因子与评价

在因子平台中，因子库功能和因子评价功能为用户提供了高效的因子开发、测试、评价和管理流程。这两个模块共同构建了一个完整的因子开发生态，确保因子从创建到应用的每一个环节都能够顺畅进行。

3.2.1 我的因子

因子库为用户提供了一个综合的环境，便于管理各种因子模块。在因子库中，用户可以通过“我的因子”查看所有现有的私有因子，并通过多种操作如编辑、测试、公开或删除进行管理。以下是因子库功能的具体细节：

- **代码编辑**：用户能够直接在平台上进行因子代码的编辑，这不仅包括添加新的逻辑，还可以修改现有算法。平台提供了语法高亮、错误提示等智能提示功能，帮助用户减少编程错误并提高开发效率。
- **因子版本管理**：因子库支持版本控制，用户可以轻松查看和恢复先前版本的因子代码。每次修改都会自动生成新的版本记录，确保因子开发过程中的每一步都可以追溯。

图3-4 因子模块编辑页面

- **测试与执行**：因子可以在界面内部即时进行测试。用户可以配置多种测试参数，选择适合的因子计算模板，便捷地启动计算过程。系统会自动生成测试日志，记录每次测试的输入参数及输出结果，以便后续分析和调整。

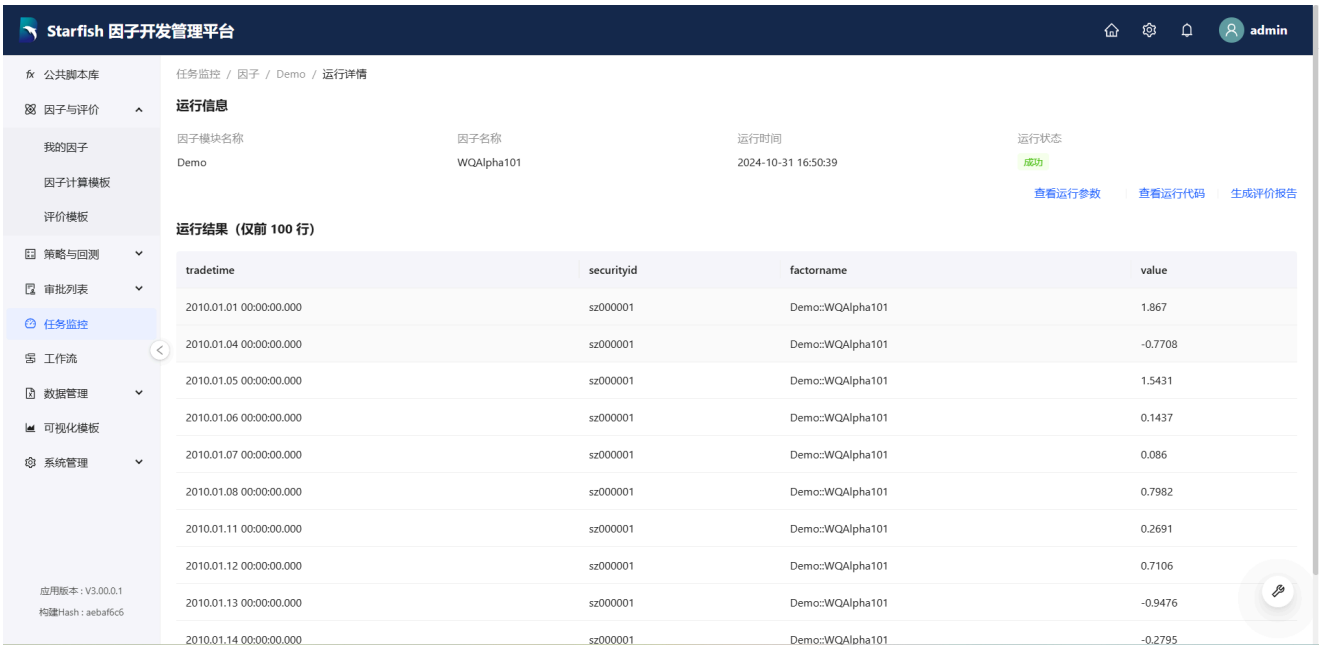

图3-5 因子测试结果页面

3.2.2 评价模板

因子评价是对因子结果进行系统评估的核心部分，用户可以在这里对因子进行深入的测试和验证。此功能模块包含以下主要特性：

- **评价模板创建与编辑**：用户可以创建和编辑因子评价模板，这些模板用于系统性地评估因子的性能和适用性。模板的灵活性允许用户根据不同需求定制评价标准，涵盖收益分析、风险评估等多种维度。
- **运行与参数配置**：在评价过程中，用户能够灵活配置各项参数，以调整评价标准和输入数据。这种灵活性确保了评价能够适应不同市场环境和投资策略，提高了因子评价的针对性和有效性。

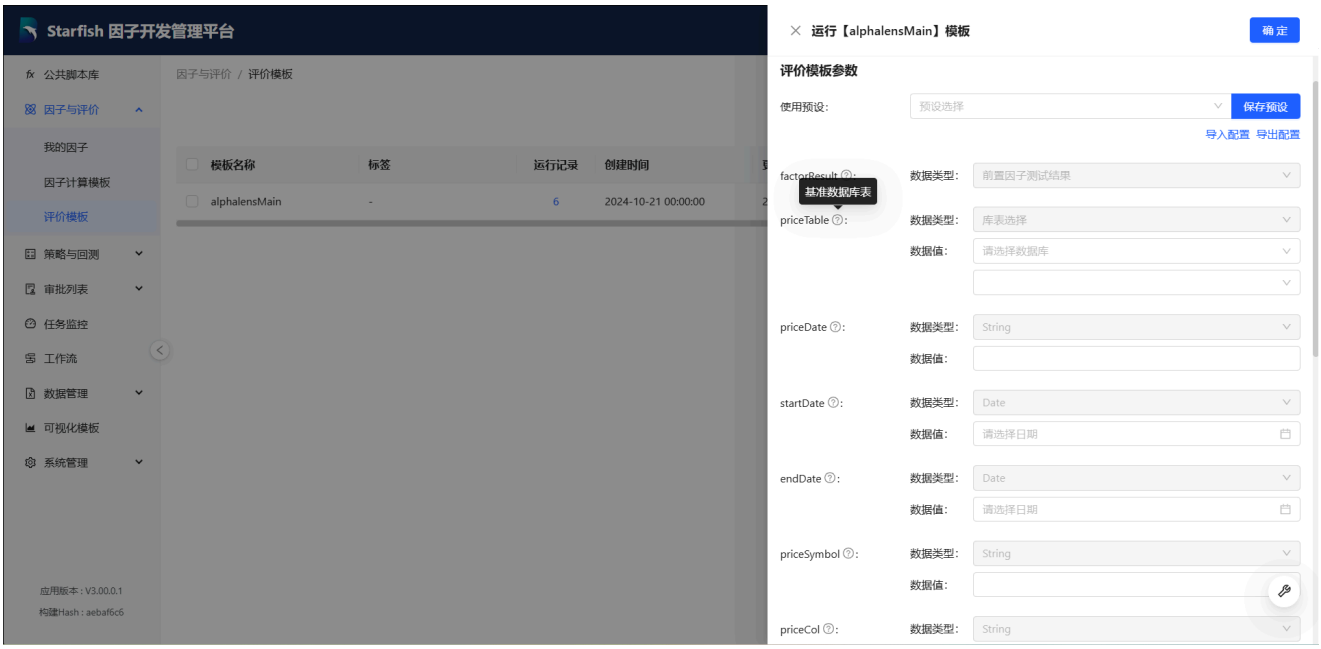

图3-6 模板运行参数设定

- **结果可视化**：评价结果可以通过内置的可视化工具进行展示，包括图表和报告生成，帮助用户直观理解因子表现。可视化工具支持多种图表类型，用户可以选择最能反映因子表现的方式，便于高效决策支持。

图3-7 Alphalens 评价可视化报告

这些功能共同支持一个完整的因子开发周期，从初步的代码编写到深入的性能评估，最终到因子的发布和应用，都能在一个统一的平台上高效进行。每个功能都被设计为支持团队协作和知识共享，同时确保代码和数据的安全性，为用户提供一个全面、可靠的因子开发环境。

3.3 策略与回测

策略功能支持用户从策略创建、回测到绩效归因的整个流程，形成一个高效的策略管理系统。这一功能集涵盖策略的管理、运行设定、执行以及结果评估，确保用户能够顺畅地开发、测试和优化投资策略。

3.3.1 我的策略

在策略库中，用户可以轻松查看所有私有策略，获取策略的详细信息，包括名称、类型、数据源、策略类型、创建时间和更新时间等。这一功能带来以下优势：

- **策略代码编写框架**：平台为用户提供了清晰的策略代码编写框架，支持标准化和模块化的开发过程。用户可以根据指导文档编写策略，利用平台提供的辅助文档进行快速上手。

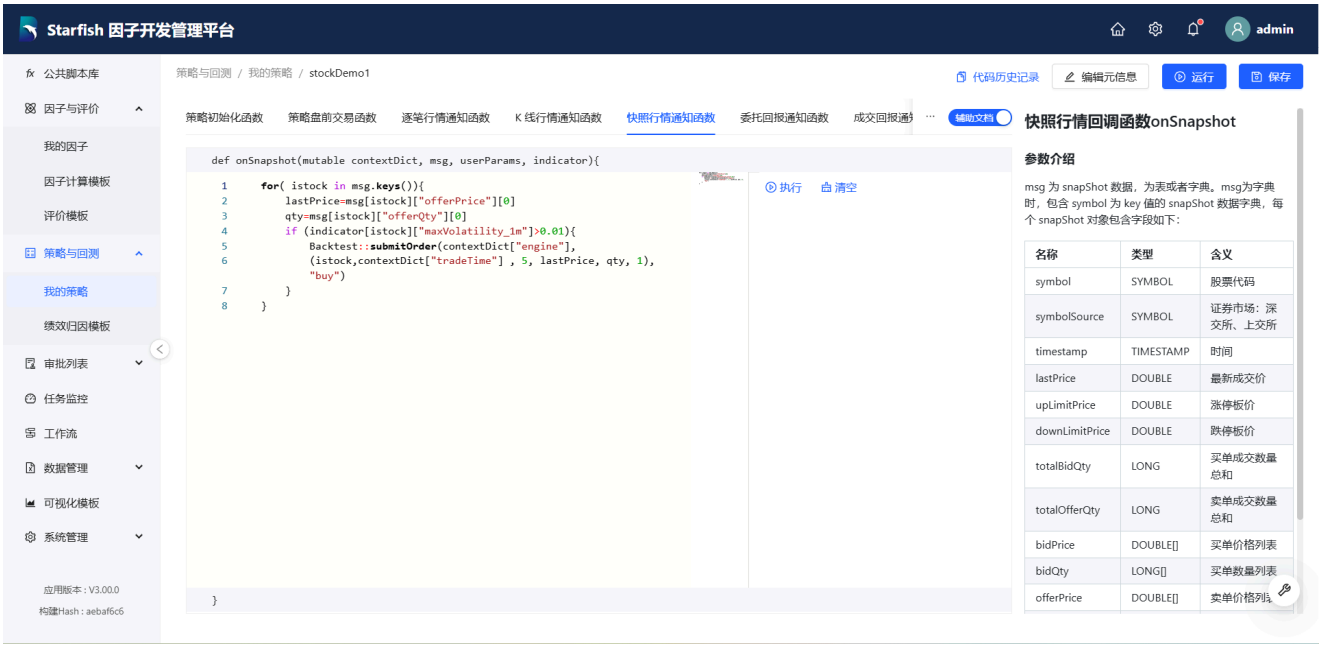

图3-8 策略编写页面

- **复制与共享**：用户可以复制现有策略，快速生成基于原有策略的修改版。这种复制功能大大加快了策略迭代的速度，用户也可以通过导入导出功能，轻松分享和迁移策略配置，促进团队间的协作。

- **回测预设功能**：为了加速策略的测试流程，平台提供回测预设功能。用户可以为相同类型的策略设置默认参数，这些预设参数可在不同回测中重复使用，并支持参数优化，帮助用户寻找最佳的策略配置。

- **结果展示**：用户可以查看每个策略的回测结果，包括收益率、最大回撤、夏普比率等关键绩效指标。结果展示提供了清晰的统计数据和图表，帮助用户快速把握策略表现。

- **可视化工具**：策略库集成了多种可视化工具，用户可以通过动态图表展示策略收益曲线、风险指标以及其他相关分析。这些可视化效果不仅提升了结果的可理解性，也增强了用户对策略的信心。

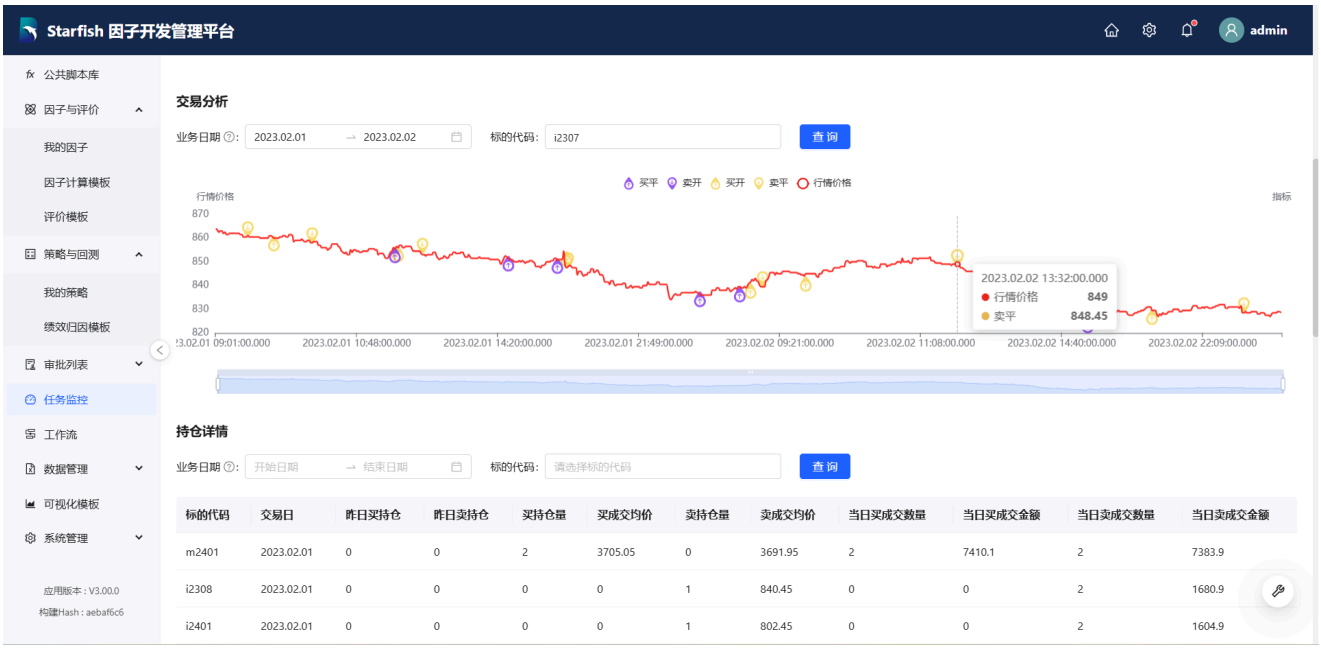

图3-9 回测结果可视化报告

3.3.2 绩效归因模板

策略回测的绩效归因功能允许用户深入分析回测结果，理解策略表现的驱动因素，具体包括以下内容：

- **自定义绩效归因模板**：用户可以根据需要创建和定制绩效归因模板，设定相关的评价参数。这种灵活性使得用户能够针对不同策略的特性，选择适当的评价指标，以获得更具针对性的分析结果。
- **可视化展示**：在评价过程中，用户可以选择合适的可视化模板，以直观展示回测结果。图表和报告将清晰呈现策略的收益构成、风险因素及市场环境影响，帮助用户快速理解策略表现。
- **深入分析**：通过绩效归因功能，用户能够更好地理解策略在不同市场条件下的表现，识别出策略成功与否的主要驱动因素。这种深度分析对于后续策略的优化和调整至关重要。

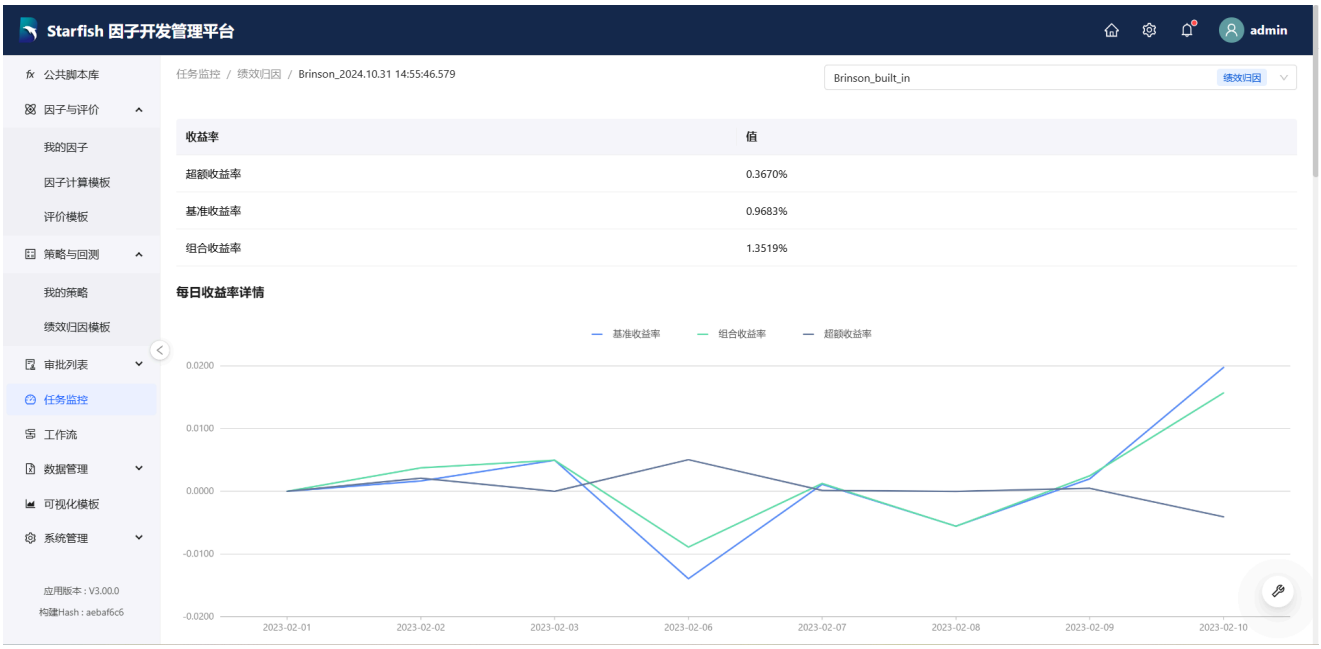

图3-10 内置 Brinson 可视化报告

3.4 审批

审批功能旨在帮助用户高效地审核因子、因子计算模板和策略的公开请求，确保团队协作中的安全性和透明度。该功能通过系统化的审批流程和严格的权限管理，使得审核过程既高效又有序。用户可以在“我的提交”页面查看自己的审批记录，执行撤回操作，或查看详细的审批信息，包括因子或策略的具体代码和相关评价报告。

审批处理：具备审批权限的用户，可以在“我的审批”页面处理其他用户的提交，支持通过或拒绝审批，并对审批意见进行反馈。这一功能增强了团队内部的协作，确保所有提交都经过适当的审查。

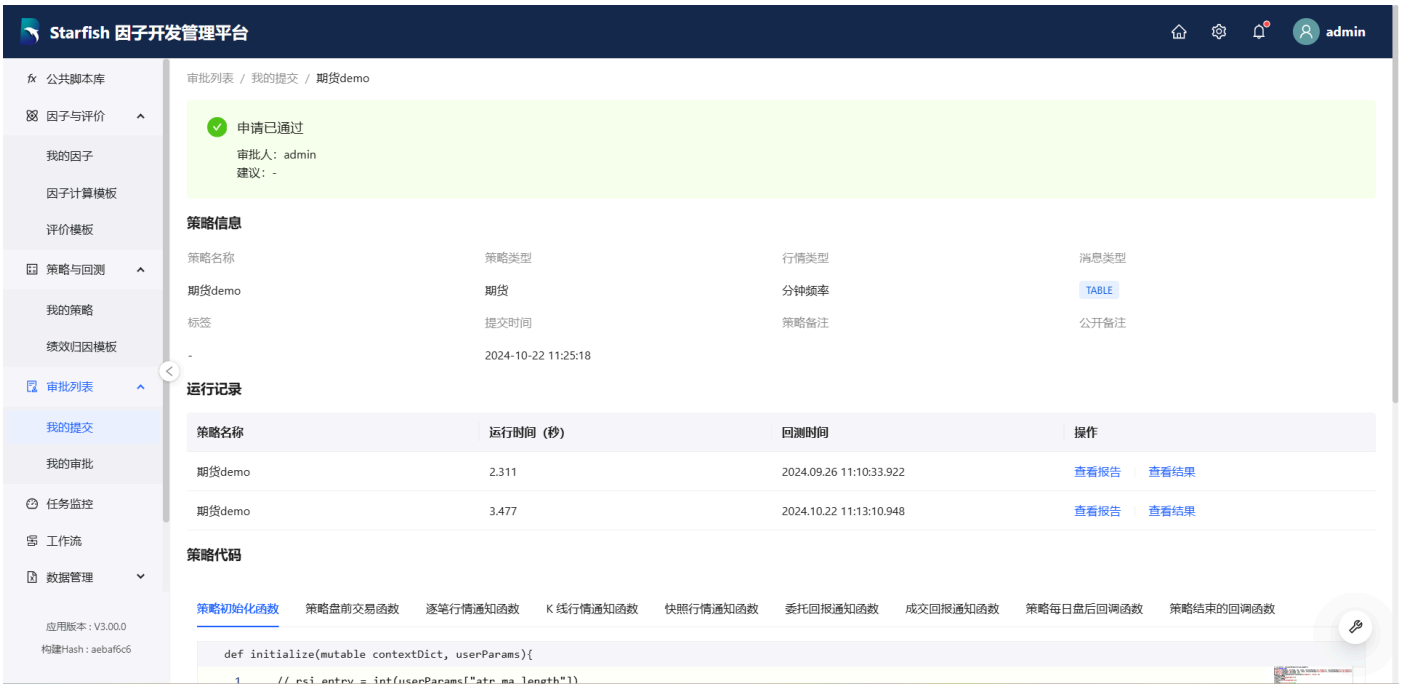

图3-11 策略的申请公开页面

通知管理：消息列表功能为用户提供了一个集中的通知区域，显示所有相关的审批活动和状态更新。用户可以快速跳转到详细审批页面，并一键标记所有消息为已读，确保审批过程的透明度与效率。

图3-12 来自审批的消息提示

整体而言，审批系统通过严格的权限管理和自动化的流程控制，为团队协作提供了必要的安全保障，使得因子和策略的发布与管理更加高效。

3.5 任务监控

任务监控是因子平台的核心功能之一，旨在监控并管理已启动的各种任务。用户和管理员可以在该界面查看任务的进展情况，包括因子测试、因子评价、策略回测、工作流、数据导入以及可视化模板等。该功能提供以下优势：

图3-13 任务监控中的策略页面

实时状态更新：用户可以实时查看各项任务的状态，访问详细信息，如运行参数、计算模板、评价结果和运行日志。这种透明性使得用户能够快速掌握任务进展并及时做出调整。

手动控制：用户具备手动终止正在进行的任务的能力，确保在必要时能够迅速响应潜在问题。

管理功能：管理员除了可以查看所有用户的任务记录外，还能进行任务过期自动驳回的管理。这一机制确保了系统资源的有效利用和任务的及时处理。

筛选与定位：任务监控模块支持任务筛选功能，用户可以快速定位特定任务，查看其详细的执行情况和结果。这种便捷性提升了用户的操作效率。

工作流管理：该模块还支持复杂的工作流管理，展示每个工作流及其子任务的运行状态和日志，确保任务执行的进度准确无误。通过有效的任务监控，用户和管理员能够全面掌握平台运作，优化整体工作流程。

图3-14 任务监控中的工作流运行状态

3.6 工作流

因子平台的工作流功能旨在串联和自动化多个任务，如数据导入、因子计算、策略回测和可视化报告生成。用户能够轻松地将这些任务组合成一个完整的工作流，实现任务之间的顺序和依赖关系。例如，用户可以设置每天收盘后自动导入当天的数据，随后进行因子值计算，并根据计算结果生成可视化报告或验证策略。这种自动化流程大大提高了效率，确保数据处理和分析任务的连续性。

可视化操作界面：用户可以在工作流模块中通过直观的拖拉拽界面编辑每个子任务，将其连接并组成完整的工作流。每个任务模块的连接方式清晰可见，用户可以根据需求调整任务顺序，直接运行或设置为定时运行。这种灵活的操作方式使得用户能够快速构建和修改工作流，满足不同的分析和决策需求。

图3-15 工作流的编辑页面

定时任务管理：工作流功能还支持定时任务的设置，允许用户在固定时间和频率下自动执行工作流，例如每日或每周触发。这确保了重要的数据处理和分析任务不会被遗漏。用户可以在定时任务管理界面中轻松调整执行时间、频率，或在需要时修改和删除已设定的任务。

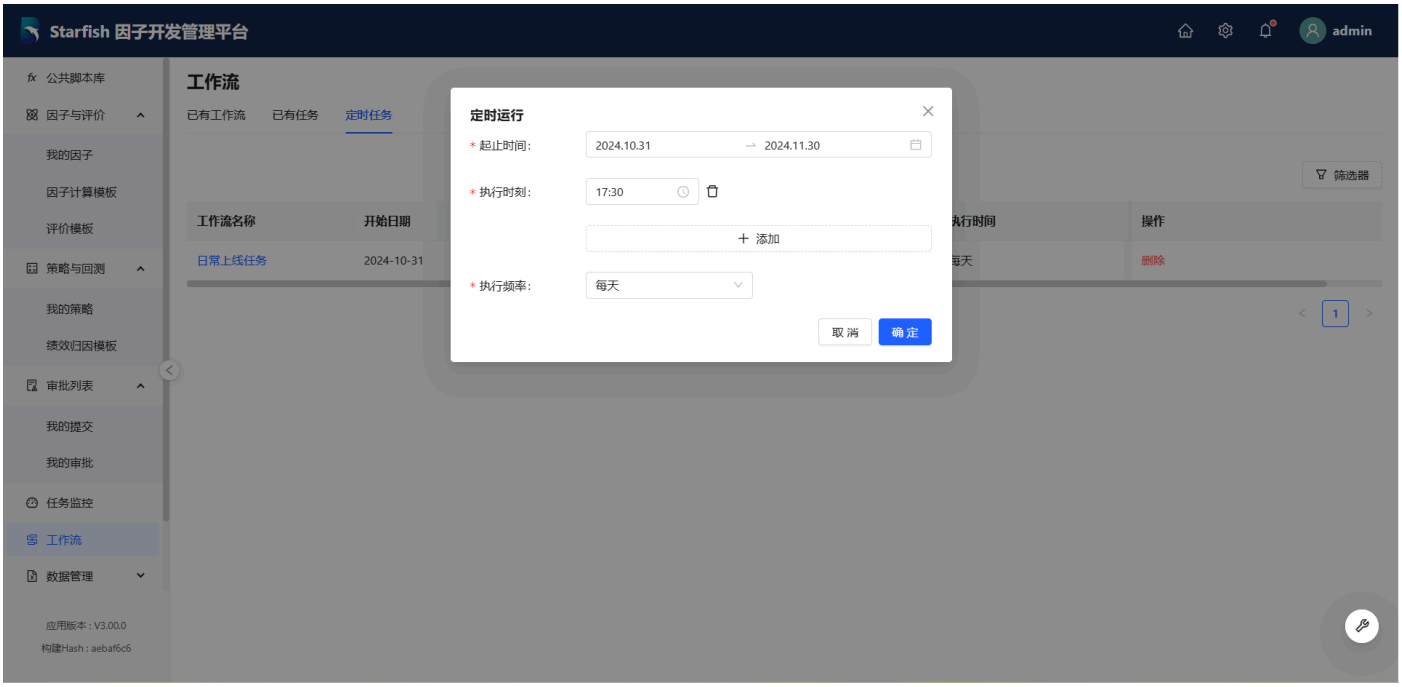

图3-16 工作流定时执行设置

因子任务设置：在工作流中的因子任务，用户还可以设置计算结果是否入库，并选择对应的因子库表。需要注意的是，入库的计算结果必须与选定库表的结构相同，以确保数据的一致性和完整性。这样的设计增强了工作流的灵活性和功能性，使得用户可以在执行任务的同时，自动管理因子数据的存储与更新。

这些功能的整合不仅增强了平台的自动化能力，也使得用户能够更有效地管理和执行复杂的因子或策略任务，从而支持高效的决策制定过程。

3.7 数据管理

因子平台的数据管理功能支持全面的库表管理、数据权限控制以及数据导入工具，旨在满足用户在数据分析和策略开发中的多样需求。该功能由库表一览和数据导入模板两个主要模块构成，使用户能够高效管理和利用平台的数据资源。

3.7.1 库表一览

库表管理：库表一览功能分为四个主要部分：可访问库表、我管理的库表、库表创建管理和赋权记录。这一结构使用户能够根据权限查看自己可以访问的库表，并管理自己负责的库表。用户可以轻松设置库表的创建权限，并查看历史赋权操作记录，以便及时了解权限变动情况。

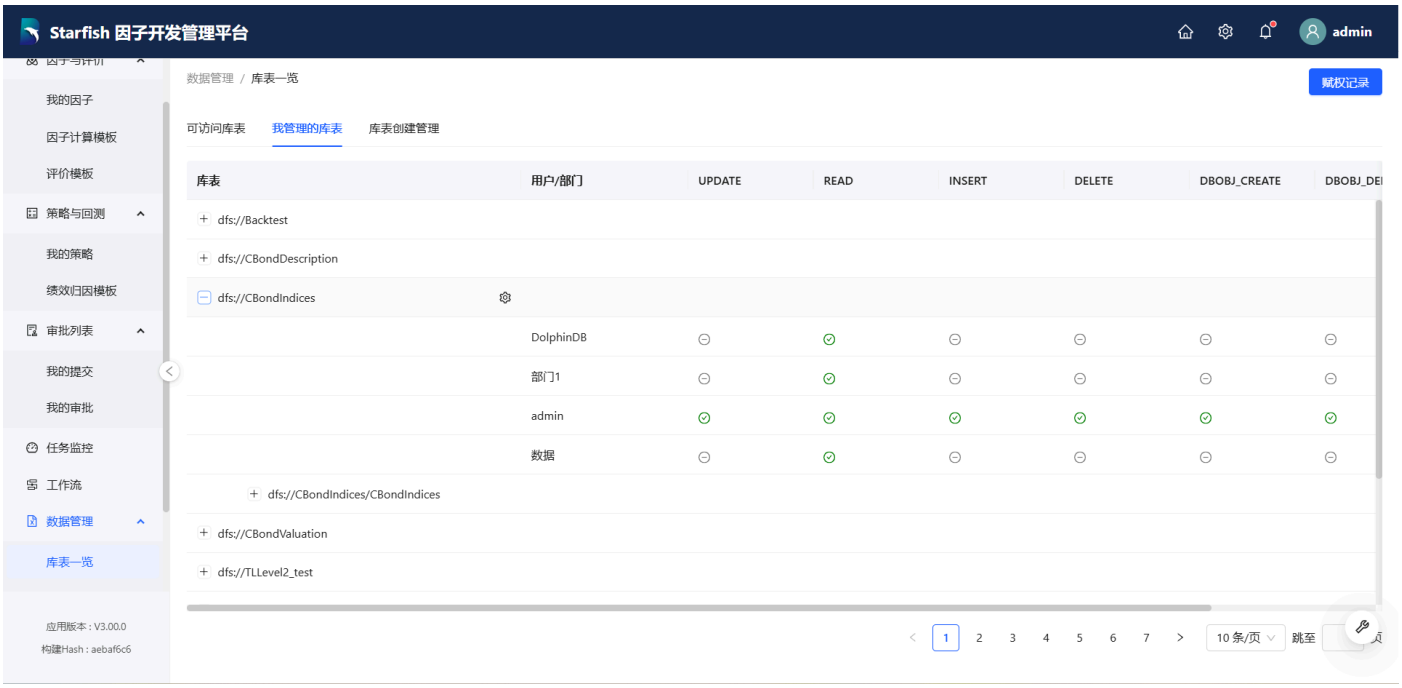

图3-17 我管理的库表页面

详细信息：在库表一览中，用户可以查看每个库表的详细信息，包括字段及其说明。这使得用户能够清晰了解数据结构，从而更好地进行数据分析和策略开发。同时，用户还可以根据权限执行赋权操作，灵活地增加或删除访问权限，确保数据管理的安全性和有效性。

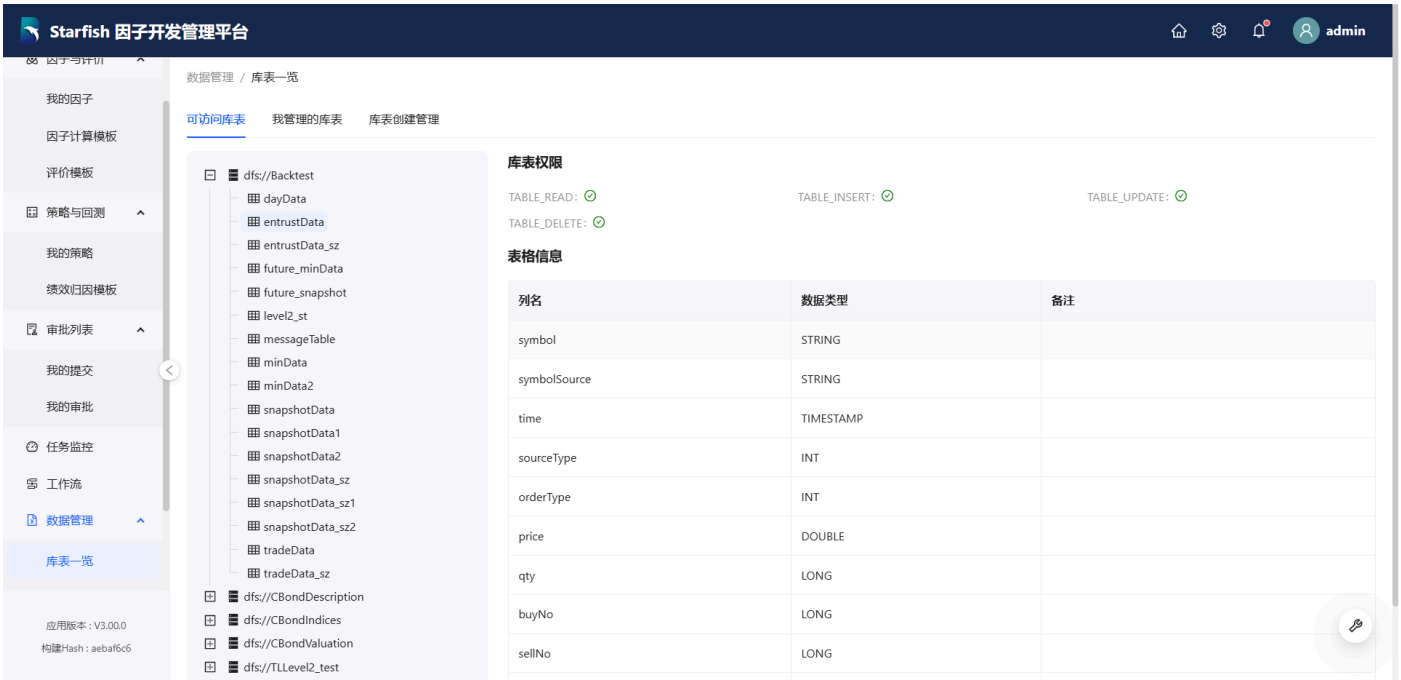

图3-18 可访问库表页面

3.7.2 数据导入模板

数据导入模板功能允许用户创建和运行数据导入脚本，从而高效地将外部数据导入到平台中。用户可以选择使用预设的导入模板或根据特定需求自定义导入模板，简化数据导入流程。用户能够实时监控数据导入任务的运行状态，并查看和管理导入模板的详细代码。这一功能确保数据导入的正确性和效率，使得用户能够高效地整合外部数据资源，支持后续的数据分析与策略开发。

3.8 可视化模板

可视化模板管理功能使用户能够轻松创建、编辑和查看多种类型的可视化模板，包括回测策略、因子计算、因子评价、绩效归因和自定义模板。这些模板旨在帮助用户将复杂数据转化为直观图形，提升理解与决策的效率。

模板创建与编辑：用户可以新建模板，选择模板类型后进入编辑页面。在编辑页面中，用户可以设定模板的基本信息，如模板名称、描述、数据源以及可视化设计元素。该界面支持拖放功能，允许用户通过简单的操作来构建所需的可视化效果。创建者对模板拥有完全的编辑权限，可以随时修改图表信息、选择数据字段和设定样式，以实现个性化的展示效果。

图3-19 可视化报告搭建页面

内置与自定义：平台还提供了一系列丰富的内置可视化模板，包含 Alphalens、Brinson、Campisi 等，用户可以直接对运行结果应用这些模板，快速获得可视化展示。此外，用户可以根据自己的需求，自由组合多个运行结果，将所有结果展示在同一可视化报告中。这种灵活性大大增强了数据分析的表现力，使得用户能够更有效地传达和分享分析结果。

这些组件的灵活组合使用户能够构建多样化的可视化报告，满足不同的分析需求，帮助用户更有效地传达和分享分析结果。此功能增强了用户的分析能力，为投资决策提供了强有力的支持。

3.9 系统管理

在因子平台中，为了高效管理用户和部门，系统提供了全面的管理工具。管理员（admin）能够轻松完成用户创建、权限配置、部门管理等一系列操作。用户也可以根据部门由相应的管理员进行分组权限管理，从而提升团队的协作效率。

用户管理：管理员可以通过用户管理页面查看所有用户的详细信息，包括所属部门、权限和账号状态。他们具备创建新用户、编辑现有用户信息、激活或禁用用户账号的权限。此外，管理员还能够调整用户的部门归属和权限配置，确保每个用户都能在合适的权限范围内工作。

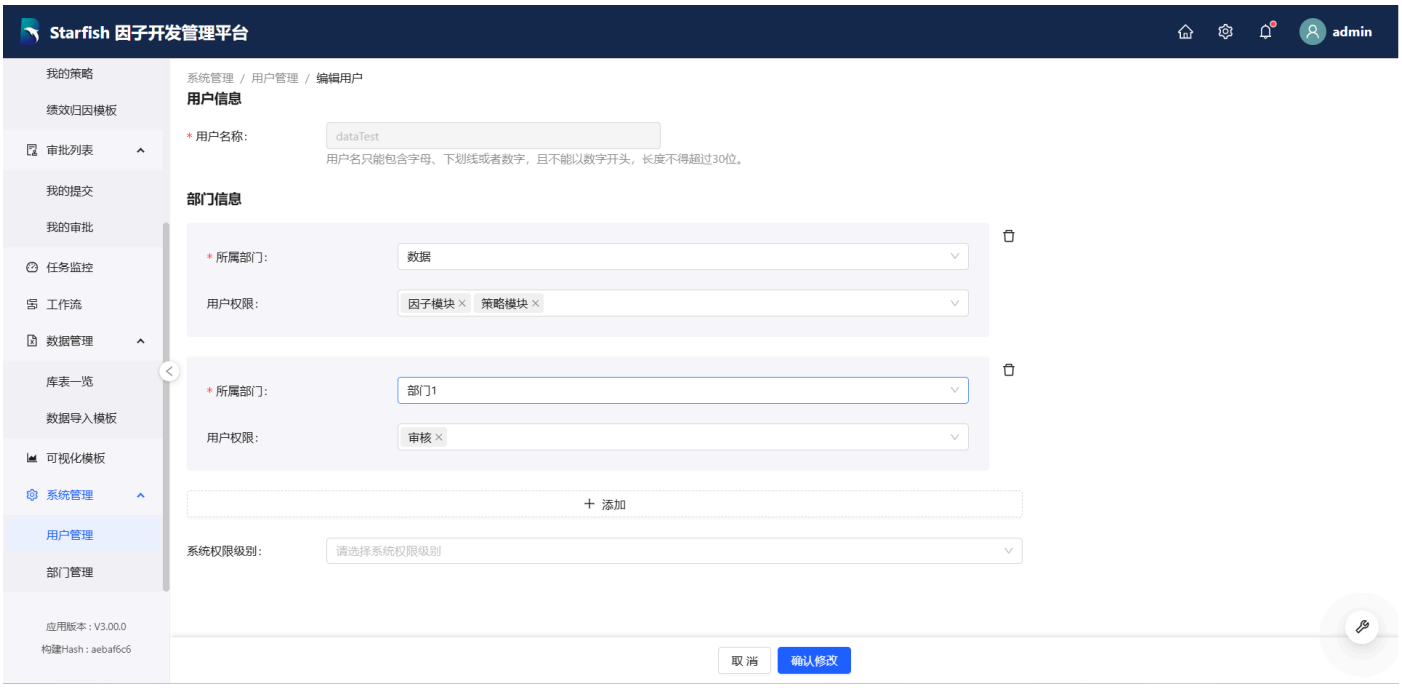

图3-20 用户编辑页面

部门管理：部门管理功能让管理员能够创建和删除部门，并设置或更改部门的管理员和成员。部门信息页面显示了部门ID、部门名称、创建时间以及部门成员和管理员的详细信息。管理员可以通过悬停在成员或管理员数字上查看具体的人员名称，便于进行部门成员的详细管理。

图3-21 部门编辑页面

这些管理功能确保了因子平台能够根据组织的结构和需求灵活调整，同时保持对关键操作的严格控制和监督。这种管理机制不仅提升了操作的透明度，还促进了团队间的有效沟通和协作，帮助用户在一个安全和高效的环境中开展工作。

第 4 章. API 接口

在这一章节中会详细介绍因子平台的 API 接口调用，包括错误码和接口规范。在因子平台中，API 调用提供了灵活的方式来实现策略的创建、管理和回测。这些接口允许用户通过编程方式与系统进行交互，从而自动化操作和流程，提升效率和准确性。通过统一的错误码格式，用户能够快速识别和处理 API 调用中的问题。同时，接口规范强调了函数名和参数名的命名约定，确保了代码的可读性与一致性。通过简洁且有意义的命名，用户可以更直观地理解各个接口的功能与用途。整体而言，API 接口的设计旨在提升用户的使用效率与开发体验，为策略的自动化与回测提供了强有力的支持。

4.1 API 功能列表

1. 因子与评价

1.1 因子模块

- starfish::facplfBasic::facplf_create_new_factor
- starfish::facplfBasic::facplf_edit_factor
- starfish::facplfBasic::facplf_get_factor_list
- starfish::facplfBasic::facplf_get_factor_detail
- starfish::facplfRun::facplf_create_draft_test
- starfish::facplfBasic::facplf_get_single_test_detail

1.2 因子计算模板

- starfish::facplfBasic::facplf_create_new_private_factor_template
- starfish::facplfBasic::facplf_edit_private_factor_template
- starfish::facplfBasic::facplf_get_private_factor_template_list
- starfish::facplfBasic::facplf_get_private_factor_template_detail

1.3 因子评价模板

- starfish::facplfBasic::facplf_create_new_analysis_template
- starfish::facplfBasic::facplf_edit_analysis_template
- starfish::facplfBasic::facplf_analysis_template_list
- starfish::facplfBasic::facplf_analysis_template_detail

- starfish::facplfRun::facplf_run_analysis
- starfish::facplfBasic::facplf_get_analysis_record_detail

2. 策略与回测

2.1 策略开发

- starfish::facplfBasic::facplf_create_new_strategy
- starfish::facplfBasic::facplf_save_strategy_code
- starfish::facplfBasic::facplf_get_strategy_list
- starfish::facplfBasic::facplf_get_backtest_detail

2.2 回测预设

- starfish::facplfBasic::facplf_create_strategy_setting
- starfish::facplfBasic::facplf_edit_strategy_setting
- starfish::facplfBasic::facplf_get_strategy_setting_list
- starfish::facplfBasic::facplf_get_strategy_setting_detail

2.3 运行策略

- starfish::facplfRun::facplf_run_backtest
- starfish::facplfRun::facplf_get_backtest_result
- starfish::facplfBasic::facplf_get_single_backtest_result
- starfish::facplfRun::facplf_run_real_time_backtest
- starfish::facplfBasic::facplf_get_real_time_backtest_stream_tbs

2.4 绩效归因

- starfish::facplfBasic::facplf_create_new_attribution_template
- starfish::facplfBasic::facplf_edit_attribution_template
- starfish::facplfBasic::facplf_attribution_template_list
- starfish::facplfBasic::facplf_attribution_template_detail

- starfish::facplfRun::facplf_run_attribution
- starfish::facplfBasic::facplf_get_attribution_record_detail

3. 数据导入模板

- starfish::facplfBasic::facplf_create_new_data_import_template
- starfish::facplfBasic::facplf_edit_data_import_template
- starfish::facplfBasic::facplf_create_new_data_import_template
- starfish::facplfBasic::facplf_data_import_template_detail
- starfish::facplfRun::facplf_run_data_import
- starfish::facplfBasic::facplf_get_data_import_record_detail

4. 工作流

- starfish::facplfBasic::facplf_get_workflow_list
- starfish::facplfRun::facplf_workflow_run

5. 任务监控

- starfish::facplfBasic::facplf_get_test_records
- starfish::facplfBasic::facplf_get_analysis_records
- starfish::facplfBasic::facplf_backtest_run_record_list
- starfish::facplfBasic::facplf_attribution_run_record_list
- starfish::facplfBasic::facplf_get_workflow_records
- starfish::facplfBasic::facplf_get_data_import_records

4.2 API 调用列举

以下是常见的 API 调用示例，涵盖了从连接服务器到执行回测的一系列步骤。通过这些 API，用户能够轻松管理策略、获取数据并执行复杂的分析任务。

在这一节中，将介绍一个 API 调用示例，展示如何通过编程接口与因子平台进行交互。

创建新因子模块 starfish::facplfBasic::facplf_create_new_factor

用途：新建因子模块。

入参：dict，包含以下键值对：

参数	类型	是否必选	说明
name	STRING	是	因子模块的名称
properties	STRING	否	因子模块的属性，多个属性可以用逗号分隔
tags	STRING	否	因子模块的标签，多个标签可以用逗号分隔
comment	STRING	否	因子模块的备注

返回：dict，包含以下键值对：

键	值类型	说明
factor_id	UUID	因子模块的唯一 id

例子：

```
starfish::facplfBasic::facplf_create_new_factor(
{"name": "apitest_factorModule","properties": "SQL,Level2","tags": "test"})
>>
factor_id->d1d3c1e8-083e-781a-6f66-839acf3e5105
```

4.3 API 接口规范

从 4.1 的 API 接口中，可以看出 API 接口遵循一定的命名和结构规范，以确保易于使用和维护。具体如下：

函数名：

- **命名规则**：所有 API 接口的函数名应以 `starfish::facplfBasic::facplf_` 或 `starfish::facplfRun::facplf_` 开头，便于分类和识别不同模块的功能。
- **简洁有意义**：函数名称应简洁且直观，清楚表达其执行的操作。例如，使用 `facplf_get_strategy_list` 来表示获取策略列表的操作，确保用户能够快速理解其功能。

参数入参：

- **结构要求**：入参可以为空或为一个字典，字典中包含键值对。具体需要的键值对在 API 手册中有详细介绍，以确保用户在调用时能够正确传递参数。
- **键值命名：**所有函数参数名应采用驼峰命名法，例如 `startDate`、`endDate`、`strategyId` 等。使用简洁且富有意义的名称，以便于理解其用途和功能。

返回值：

- **数据结构**：API 的返回值没有固定的数据结构，但大部分返回字典格式。具体的返回数据结构和内容应在 API 手册中详细说明，确保用户在接收返回值时能够正确解析。
- **错误处理**：API 返回的字典应包含错误码和描述信息，以便用户能快速识别问题并进行调试。例如，返回格式可以为 `{"code": "S001", "message": "不具备该权限"}`。

4.4 错误码

API 接口捕获的报错，均用统一格式的错误码输出，例如执行查看一个草稿策略的详情，

```
starfish::facplfBasic::facplf_get_backtest_detail(
  {id: "10000000-0000-0000-0000-000000000000"})
```

由于但是传入了一个错误的策略 id 后，返回了如下信息：

```
starfish::facplfBasic::starfish::facplfBasic::facplf_get_backtest_detail(
dictf0ba4e1e097f0000) => starfish::facplfBasic::facplf_get_backtest_detail:
throw toStdJson(dictf0ba4e1e097f0000) => {"code": "S105"}
```

此处的” S105 “即为错误码。现有错误码对应的报错有：

```
S001: 不具备该权限
S002: 该用户已被禁用或删除
S003: 该用户已被激活
S004: 该用户不存在
S005: 该用户已存在
S006: 无法删除该部门
S007: '{{variables}}用户已被禁用或删除'
S008: 公开数据权限赋予失败
S009: 公开审核权限赋予失败
S010: 该部门昵称已存在
S011: 该部门名称已存在
S012: 部门创建失败
S013: 因子模块草稿不存在
S014: 因子模块草稿名称已存在
S015: 无法找到因子模板代码
S016: 库表创建失败
S017: 检测失败，因子值数据有误
S018: 因子模板草稿不存在
S019: 因子模板草稿名称已存在
S020: 部门审核员不存在
S021: 该用户未加入部门
S022: 传入参数格式错误
S023: 该用户为部门管理员，无法禁用或删除
S024: 撤回已发布因子模块失败
S025: '创建用户失败，报错原因：{{variables}}'
S026: 审批失败
```

```
S027: 保存失败，函数名需与模板名称保持一致
S028: 代码不能为空
S029: '该公开{{variables}}不存在'
S030: '编辑失败，{{variables}}正在审核中，请先撤销审核'
S031: 已读该条消息
S032: 无已读消息
S033: 找不到该审批记录
S034: 该测试不存在
S035: 不允许删除该用户
S036: 该因子模块不存在或不允许访问
S037: 工作流不存在或工作流为空
S038: 该工作流当前无运行记录
S039: 该工作流当前无定时任务设置
S040: '该工作流存在定时任务，若需要修改工作流内容，请先删除定时任务'
S041: 该工作流当前无运行记录详情
S042: 任务已结束运行
S043: 部门组删除失败
S044: 创建部门失败，管理员不能为空
S045: 该工作流任务不存在
S046: '操作失败，该名称已存在公开{{variables}}中'
S047: '该代码执行有错误，{{variables}}'
S048: 部门选择错误
S049: '删除失败，{{variables}}正在审核中，请先撤销审核'
S050: '以下因子模块名称{{variables}}与已公开因子库中模块重名，请改名后再发布'
S051: 删除权限失败，库表不存在
S052: 该权限赋予失败，权限不存在
S053: 创建部门失败，管理员和成员不能重复
S054: 该函数库名称已存在
S055: 该函数库执行有错误
S056: 该函数库不存在
S057: 无操作权限
S058: 该数据导入模板名称已存在
S059: 该数据导入模板不存在
S060: 该数据导入函数模板执行有错误
S061: 该部门不存在
S062: 该数据导入任务不存在
S063: 删除权限和增加权限有重复
S064: 权限错误
S065: 该权限已存在
S066: 用户未加入当前部门，无法修改
S067: 用户已加入当前部门
S068: 该库表不存在当前部门
S069: 库表修改失败
S070: 用户已拥有当前库表访问权限
S071: 用户没有当前库表访问权限，删除失败
S072: 无法找到部门
S073: 该任务不存在
S074: '用户{{variables}}不存在'
```

```
S075: 传参错误
S076: 找不到该工作流的起始节点或结束节点
S077: 后台任务缓存已经被清理
S078: 任务执行时间过长，已取消任务
S079: 检测失败，获取因子测试结果错误
S080: 密码有误，修改密码失败
S081: 权限丢失，请重新赋予权限
S082: 获取用户信息失败
S083: '公开失败， 当前{{variables}}正在审批流程中'
S084: 该工作流运行记录不存在
S085: 您暂无平台使用权限，请联系管理员
S086: 该因子评价模板名称已存在
S087: 该因子评价模板不存在
S088: 该因子评价记录不存在
S089: '参数{{variables}}输入格式错误'
S090: '无法访问库表{{variables}}'
S091: '当前名称{{variables}}与DolphinDB内置函数重名'
S092: 模板代码为空，请输入正确代码后再保存
S093: 暂无因子开发管理平台权限，请管理员激活后重试
S094: 登陆失败
S095: '修改用户信息失败，{{variables}}管理员不能为空'
S096: '因子计算任务发送失败，错误：{{variables}}'
S097: 因子计算任务发送长期未响应，任务终止
S098: 定时任务删除失败
S099: 存在同名的任务或工作流，保存失败
S100:
S101: 该因子计算模板名称与已有公开因子计算模板名称重复
S102: 标签修改失败，请检查标签是否以英文逗号分隔
S103: 被复制模板的代码不能为空
S104: 该策略草稿名称已存在
S105: 策略草稿已被删除
S106: 该预设名称已存在
S107: 策略设置已被删除
S108: 策略运行记录已被删除
S109: 可视化模板名称已存在
S110: 该可视化模板不存在
S111: '可视化模板{{variables}}不存在'
S112: '对可视化模板{{variables}}无删除权限'
S113: 该策略名称与已有公开策略名称重复
S114: 该数据导入记录不存在
S115: 该处理结果得到的数据形式非字典，无法进行可视化配置
S116: 自定义可视化中代码不能为空，请返回字典
S117: 该绩效归因模板名称已存在
S118: 该绩效归因模板不存在
S119: 绩效归因结果必须是字典
S120: 无法选择当前运行任务
S121: 未检测到回测插件，本次运行失败
S122: 格式错误，导入数据失败
```

```
S123：债券类型回测暂不支持除快照外行情类型
S124：无权限查看此实时回测结果
S125：任务类型错误
S126：该节点无使用starfish权限，请检查license或咨询技术支持
S127：实时回测暂不支持除期货外其他策略类型
S128：实时回测任务数量超过限制，每人最多创建5个实时任务，请先中止或删除已有任务
```

其中{{variables}}中会包含错误的信息变量。例如，重复的策略名称等。

第 5 章. 场景案例

在因子平台的使用过程中，用户可以通过两种主要方式进行策略创建和回测：一是使用 Python 调用 API 进行自动化操作，二是通过因子平台的网页应用进行交互式创建和管理。这两种方法各有优势，适用于不同的使用场景。

通过 API 调用，用户可以实现策略的快速创建和批量处理，适合需要高效管理多个策略的用户。在这一部分，将详细介绍如何使用 Python 脚本连接到 DolphinDB 服务器，创建和编辑策略代码，并执行回测操作，获取回测结果。API 的灵活性允许用户自定义策略逻辑和回测参数，从而更好地满足个性化需求。

另一方面，通过网页应用，用户可以直观地进行策略的创建和回测。该方法适合对编程不太熟悉的用户或希望快速试验策略的人士。在网页界面中，用户可以通过图形化的界面选择策略、编辑代码、设置回测参数，并实时查看回测结果和可视化报告。这种交互式的方式不仅降低了使用门槛，也提升了用户体验。

接下来将分别深入探讨这两种方法的具体操作流程和最佳实践，帮助用户根据自己的需求选择合适的策略创建和回测方式。

5.1 使用 python 调用 api 进行回测

使用API来自动化回测过程可以极大地提高效率和灵活性。本节将介绍如何通过 Python 调用 DolphinDB 的 API 来完成策略的回测操作。具体步骤包括连接到 DolphinDB 服务器、创建新策略、编写策略代码以及执行回测，并获取结果。

在开始之前，需要确保已安装 `dolphindb` 库，以便与 DolphinDB 进行交互。接下来，，建立与DolphinDB服务器的连接。

```
import dolphindb as ddb
# 创建DolphinDB会话并连接到服务器
s = ddb.session()
s.connect(host="localhost", port=8848, userid="admin", password="123456")
```

完成连接后，可以调用因子平台的相关功能，如获取已有策略列表、创建新策略、编辑策略代码等。接下来的步骤将详细展示如何通过API进行这些操作，确保策略回测的高效和准确。

```
s.run("use starfish::facplfRun")  # 使用因子平台运行模块中的函数
# 查看已有策略列表
# 返回值为包含策略列表的字：items 中为策略list，total 为策略总数
strategy_list = s.run('starfish::facplfBasic::facplf_get_strategy_list()')
# 打印当前用户下所有策略
print(strategy_list["items"])


# 新建股票策略，数据类型为快照，以字典形式接受行情数据，备注和标签为空
# 返回值为包含策略id的字典
strategy_id_dict = s.run('starfish::facplfBasic::facplf_create_new_strategy(
                  {"strategy_name": "API_demo",
                  "strategy_group": "stock",
```

```
                  "data_type": 1, "msg_as_table": false,
                  "tags": "", "comment": ""})')
print(strategy_id_dict)
# output: {'id': '39cd8c97-4a2d-13ab-da23-4f4450e6b4aa'}

# 查看策略
# 返回值为包含策略详情的字典
strategy_detail = s.run('starfish::facplfBasic::facplf_get_backtest_detail',
                        strategy_id_dict)



# 编辑/更新策略代码
# 无返回值
#
initialize = '''
 //初始化回调函数
 print("initialize")
 //订阅快照行情的指标
 d=dict(STRING,ANY)
 d["timestamp"] = <timestamp>
 d["maxVolatility_1m"]=<lastPrice\prevClosePrice-1>
 Backtest::subscribeIndicator(contextDict["engine"], "snapshot", d)
'''

beforeTrading = '''
    ////每日盘前回调函数
    ////1、通过contextDict["tradeDate"]可以获取当日;
    print ("beforeTrading: "+contextDict["tradeDate"])
    //通过backtest::setUniverse可以更换当日股票池
    Backtest::setUniverse(contextDict["engine"],["000001.XSHE"])
'''
finalize = ''
afterTrading = ''
onBar = ''
onOrder = ''
onSnapshot = '''
 for( istock in msg.keys()){
  lastPrice=msg[istock]["offerPrice"][0]
  qty=msg[istock]["offerQty"][0]
  if (indicator[istock]["maxVolatility_1m"]>0.01){
   Backtest::submitOrder(contextDict["engine"],
   (istock,contextDict["tradeTime"] , 5, lastPrice, qty, 1),"buy")
  }
 }
'''
onTick = ''
onTrade = ''
code = {"initialize": initialize,"beforeTrading": beforeTrading,"onTick": onTick,
```

```
        "onBar": onBar,"onSnapshot": onSnapshot,"onOrder": onOrder,
        "onTrade": onTrade,"afterTrading": afterTrading,"finalize": finalize}
s.run('starfish::facplfBasic::facplf_save_strategy_code',
  {"id": strategy_id_dict["id"], "code": code})


########################
### 以下为历史回测部分 ###
########################

# 设置回测参数
msg_tb_code = '''
colName=["symbol","symbolSource","timestamp","lastPrice","upLimitPrice",
        "downLimitPrice","totalBidQty","totalOfferQty","bidPrice",
        "bidQty","offerPrice","offerQty","signal","prevClosePrice"]
colType= ["STRING","STRING","TIMESTAMP","DOUBLE","DOUBLE","DOUBLE","LONG",
  "LONG","DOUBLE[]","LONG[]","DOUBLE[]","LONG[]","DOUBLE[]","DOUBLE"]
messageTable=table(10000000:0, colName, colType)

insert into messageTable values("000001.XSHE","XSHE",
2022.04.11 10:10:00.000,7,15,5,10000,10000,
arrayVector([10],[6.9, 6.8, 6.7, 6.6, 6.5, 6.4, 6.3, 6.2, 6.1,6.0]),
arrayVector([10],[800,900,1000,1100,1200,1000,1000,1000,1000,1000]),
arrayVector([10],[7.1,7.2,7.3,7.4,7.5,7.6,7.7,7.8,7.9,8.0]),
arrayVector([10],[1000,1000,1000,1000,1000,1000,1000,1000,1000,1000]),,6.5)
insert into messageTable values("000001.XSHE","XSHE",
2022.04.11 10:10:03.000,7.5,15,5,10000,10000,
arrayVector([10],[7.4, 6.8, 6.7, 6.6, 6.5, 6.4, 6.3, 6.2, 6.1,6.0]),
arrayVector([10],[800,900,1000,1100,1200,1000,1000,1000,1000,1000]),
arrayVector([10],[7.6,7.7,7.8,7.9,8.0,8.1,8.2,8.3,8.4,8.5]),
arrayVector([10],[1000,1000,1000,1000,1000,1000,1000,1000,1000,1000]),,7)

result = messageTable
'''
config = {"date": [["2022.04.11","2022.04.11"]],"cash": [1000000],
        "commission": [0.00015],"tax": [0.001],"matchingMode": [],"benchmark": [],
        "frequency": [0],"latency": [],"symbolList": [],"msgPartition": [],
        "orderBookMatchingRatio": [],"matchingRatio": [],
        "enableSubscriptionToTickQuotes": [],"outputQueuePosition": []}
config["msgTables"] = [{"configure": True,"tableName": "snapshot",
        "mode": "code","code": msg_tb_code}]

# 设置运行参数
param = {}
param["id"] = strategy_id_dict["id"]
param["type"] = 0
param["config"] = config
# 运行历史回测
```

```
backtest_run_id = s.run('starfish::facplfRun::facplf_run_backtest',param)


# 查看回测结果
# 返回值为包含回测结果的字典，
# 包含回测任务id、策略id、回测开始时间、回测时间、回测状态、回测子任务job id、回测日志
backtest_result = s.run('starfish::facplfRun::facplf_get_backtest_result',
                        backtest_run_id)
# 查看回测子任务列表
single_job_list = backtest_result["jobs"]
# 获取回测子任务id
single_job_ids = single_job_list["job_id"].tolist()

# 例如，查看第一个子任务的回测结果
single_job_result = s.run(
                    'starfish::facplfBasic::facplf_get_single_backtest_result',
                    {"job_id": single_job_ids[0]})
# 查看回测的结果，为一个字典，包含回测结果的详细信息，分别是
# getReturnSummary，getDailyTotalPortfolios,getDailyPosition，
# getTradeDetails，getPosition，
# 分别对应策略的收益概述，策略每日权益指标，每日持仓数据详情，交易明细表，持仓信息
single_job_result_detail = single_job_result["result"]
```

5.2 使用网页应用进行回测

假设现在，我们需要在因子平台的前台网页端内创建一个股票快照策略。

在策略与回测页面下我的策略中，可以创建新策略，选择相应的策略、行情、消息类型后就会跳转此策略的编辑页面。

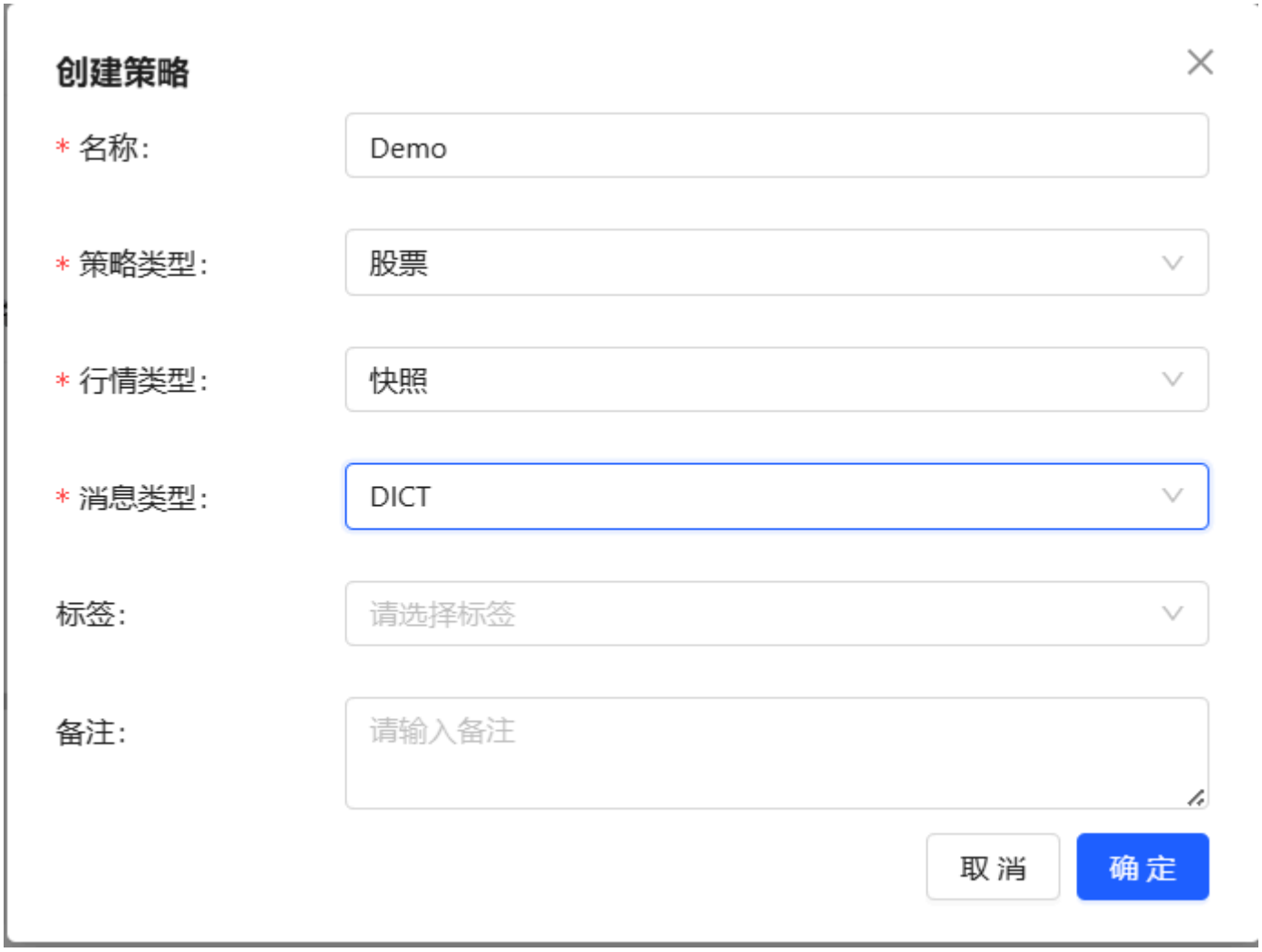

图5-1 创建策略

在编辑页面中，有 7 个函数可以进行编辑保存。

在策略初始化函数中，此策略定义了一个指标 maxVolatility_1m，为当前 tick 相比于上一个 tick 的收益率。subscribeIndicator 接口获取回测引擎名、需要计算的数据类型（如 snapshot、tick 等）、需要计算的指标字典（key为指标名，用于之后访问；value为指标计算的元代码），之后计算结果将传入 onSnapshot 等策略回调函数。

图5-2 策略编辑页面-初始化

在每日盘前回调交易函数中，可以初始化当日的全局变量并使用 Backtest::setUniverse 订阅设定了当日的股票池。

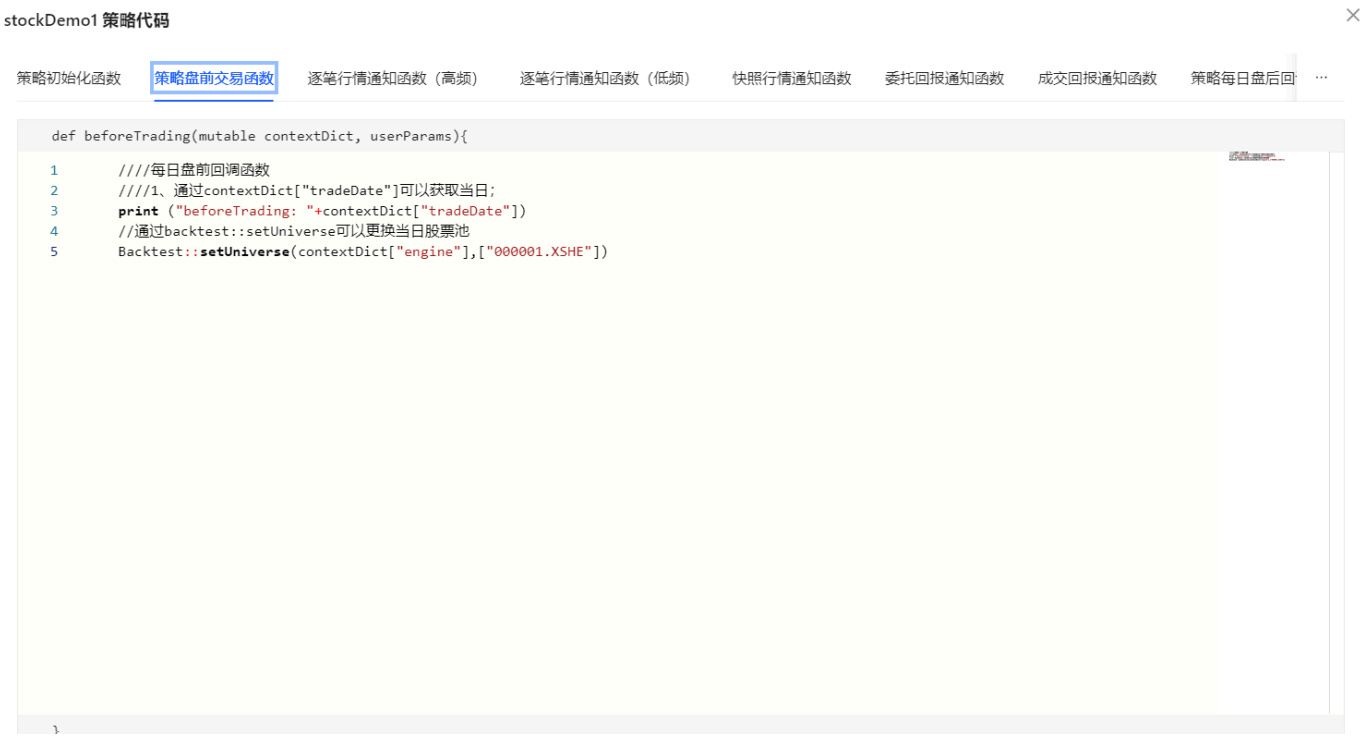

图5-3 策略编辑页面-策略盘前交易函数

在快照行情通知函数中，编写基于快照行情的策略逻辑。可以根据行情及策略信号下达委托买卖订单。onSnapshot 中的 msg 参数中包含了最新的快照行情以及在初始化中订阅的指标。例如，在 Demo 中，我们可以计算出的 maxVolatility_1m 值来判断是否需要下达委托买的订单。Backtest::submitOrder 是回测引擎提供的下单接口。

图5-4 策略编辑页面-快照行情通知函数

因此本策略的主要逻辑为：如果最新价相比于前价涨幅超过1%，则下单 000001.XSHE 的限价买单，下单时间为最新的快照时间，价格为一档卖价，股数为一档卖量。

在编辑页面或者列表点击运行按钮后，可以填写回测引擎所需的参数或者选择已经设定好的预设参数，在这里，假设我们只对 2022.04.11 这一天进行回测，并且设置一些基础信息后点击下一步。

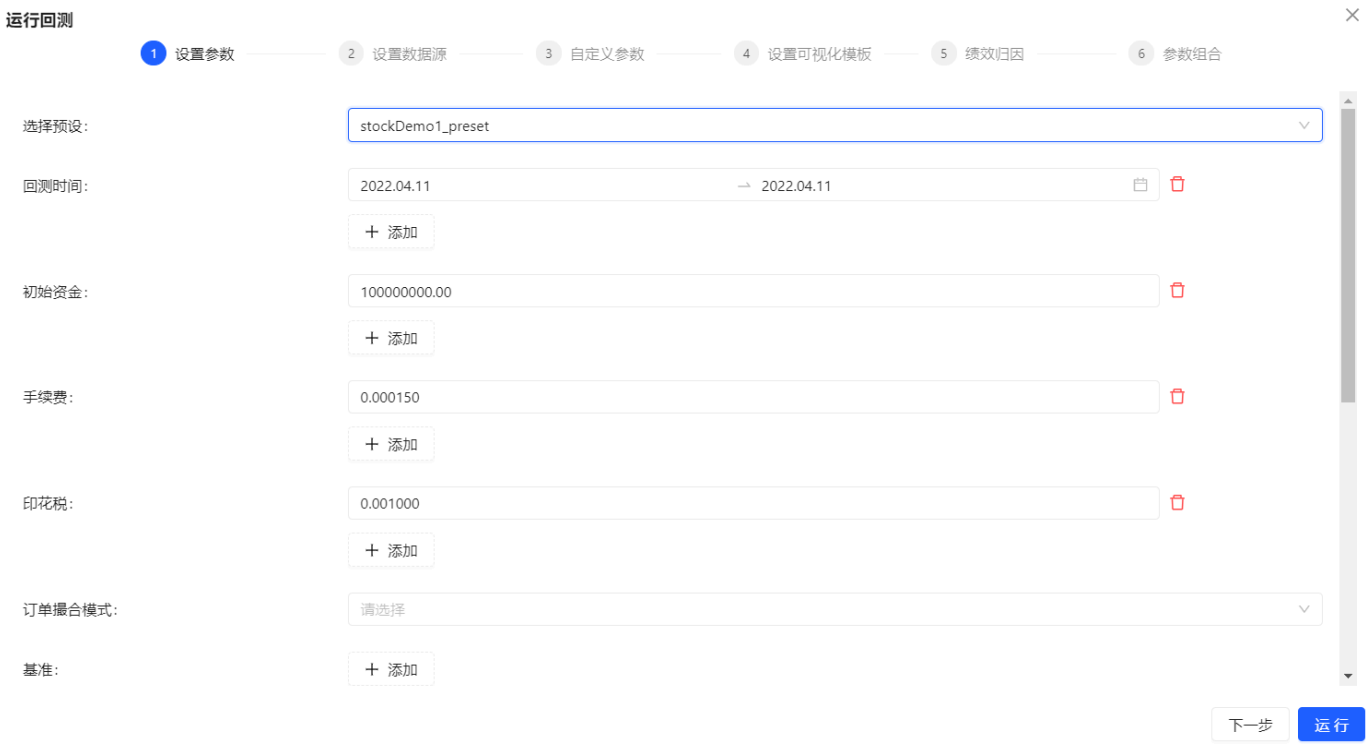

图5-5 策略回测预设页面 - 设置参数

下一步为设置数据源，包含行情数据源。在这里，我们使用的是两条模拟的 tick 数据。

图5-6 策略回测预设页面 - 设置数据源

设置完成后，所有回测必须的参数就设定完了。其余的步骤和参数为额外功能，可以具体参考快速手册或者教程进行更高阶的回测运行。点击运行即可进入运行的详情页面。当然，如果已经设置好所有所需的参数，可以在任一步骤中点击运行。回测的结果页面可以分为两个部分：

图5-7 回测结果详情页面

- 运行信息：包含此运行的元信息以及运行状态、运行参数
- 策略子任务：包含此运行中所有回测任务的运行时间、回测时间以及运行状态。

◦ 例如，在这里可以点击查看日志查看策略中打印的日志

图5-8 回测结果详情页面- 查看日志

- 点击查看详情，即可查看回测任务的详情

◦ 运行结果：里面包含了所有回测引擎返回的表格结果。可以对每个表格进行下载。

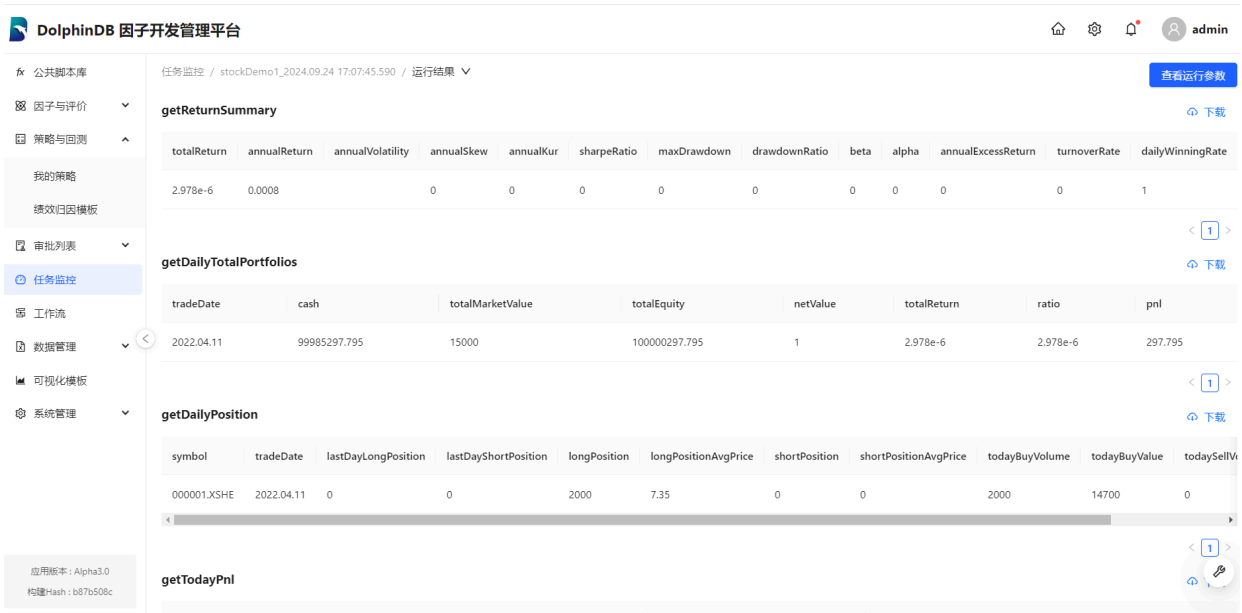

图5-9 回测结果数据报告

- 可视化报告：因为在参数处未选择模板，生成的报告使用了默认的可视化模板，包含了策略概述、收益详情、交易分析、持仓详情、账户详情和订单详情，对于后 4 个模块，可以进行股票代码和日期的筛选。

第 6 章. Roadmap

未来的因子平台开发将围绕用户体验、性能提升和功能扩展等方面进行，计划分为以下几个部分，以逐步构建更强大和灵活的因子管理与策略回测平台。

更丰富的可视化

变量支持的筛选功能：在数据预处理阶段加入变量支持，用户可以在可视化报告中动态地根据不同变量进行筛选和分析。例如，通过选择特定的因子值区间或时间窗口，用户可以快速调整分析视角，实现更灵活的数据洞察。

新增 K 线图等组件：新增更适用于金融时间序列分析的 K 线图组件，以提供清晰的历史行情展示。此外，还将引入其他可视化组件，如趋势线、热图和关联矩阵，帮助用户更直观地理解因子表现与策略收益关系。

多集群的因子平台

多集群架构：实现多集群环境下的因子平台部署，支持在不同集群间共享和管理因子数据及策略，便于大规模集成与高效运算。通过多集群架构，因子计算可以在多地分布式集群上并行进行，显著提升计算效率。

跨集群资源调度：在多集群架构下引入资源调度功能，根据负载情况灵活调配计算资源，从而优化平台的使用效率。平台将支持跨集群的因子和策略访问，实现更高效的分布式计算和数据共享。

更灵活的主页面配置

页面布局定制：为用户提供个性化的页面布局工具，使其能够选择并排列因子、策略、运行信息等模块。用户可以根据个人工作需求自定义信息展示顺序，打造更符合自身需求的操作界面。

因子、策略及运行信息展示：在主页面中，可以快速获取当前重要因子、策略运行状态、以及回测进度。平台将增加简化版的信息展示和提醒功能，便于用户即时掌握平台内的关键动态。

模块扩展和小组件配置：未来主页面将支持更多小组件扩展，例如快速访问的最近使用因子、策略运行历史等。用户可以在主页面增加和调整这些模块，快速查看并管理重点项目，提升整体效率。