

项目介绍

Kaleido AI智能衣柜是一个基于微服务架构和的AI智能衣柜管理系统。项目旨在通过AI技术帮助用户更好地管理个人衣物、创建个性化穿搭方案，并提供智能推荐服务。系统结合了现代微服务架构、LLM(大语言模型)、向量数据库、MCP等技术，为用户提供全方位的衣柜管理解决方案。

项目背景

随着人们生活水平的提高和消费观念的转变，个人衣物数量不断增加，如何高效管理衣柜、合理搭配服装成为许多人的痛点。传统的手工记录和记忆方式已无法满足现代人的需求。Kaleido项目应运而生，通过技术手段解决以下核心问题：

- 衣物管理数字化**：将实体衣物转化为数字资产，实现分类、标签、搜索等数字化管理
- 穿搭方案智能化**：基于AI为用户推荐合适的穿搭组合
- 穿着记录分析**：记录衣物穿着历史，分析使用频率和搭配效果
- 个性化推荐**：根据用户偏好、天气、场合等因素推荐穿搭

目标用户群体

- 时尚爱好者**：需要管理大量衣物并寻求专业搭配建议的用户
- 商务人士**：需要根据不同场合快速选择合适的职业装束
- 普通消费者**：希望提高衣物利用率，避免重复购买相似物品
- 服装零售商**：可作为增值服务提供给客户，增强客户粘性
- 时尚博主/设计师**：需要管理样品和创作灵感库

架构设计

Kaleido项目采用DDD（领域驱动设计）+ CQRS（命令查询职责分离）+ 整洁架构（Clean Architecture）的设计模式，确保代码结构清晰、职责明确、易于维护和扩展。

代码分层设计

项目采用四层架构设计，各层职责明确，依赖方向从外向内：

- Trigger层（接口层）**
 - Controller**：HTTP REST API控制器
 - RPC**：Dubbo RPC服务实现
 - Event**：领域事件处理器
 - Job**：定时任务处理器
 - 职责**：基础参数校验、请求转发、协议适配，不包含业务逻辑
- Application层（应用层）**
 - Command**：命令处理服务，负责写操作的业务编排
 - Query**：查询服务，实现CQRS的查询端

- **职责**：业务编排、操作日志记录、协调领域服务完成业务操作，不包含业务逻辑与参数校验
3. **Domain层（领域层）**
- **Model**：领域模型（聚合根、实体、值对象）
 - **Service**：领域服务，包含参数校验与聚合根的修改
 - **Adapter**：领域适配器
 - **Event**：领域事件定义
 - **职责**：核心业务逻辑实现，包含最核心的业务实现
4. **Infrastructure层（基础设施层）**
- **DAO**：数据访问对象
 - **Adapter**：基础设施适配器（数据库、缓存、消息队列等）
 - **Config**：配置类
 - **职责**：数据库增删改查、缓存、消息队列

整洁架构原则

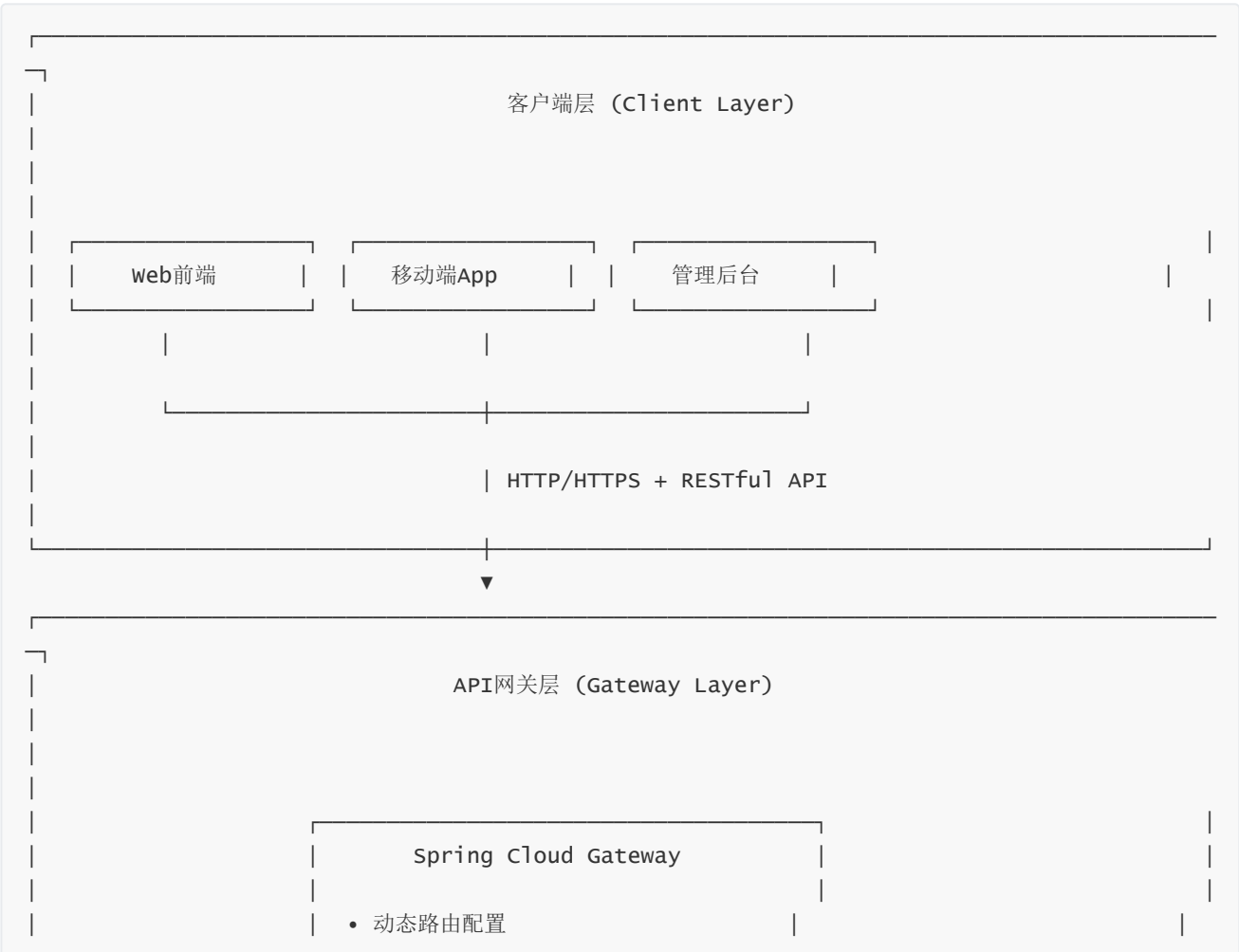
遵循整洁架构的依赖规则：

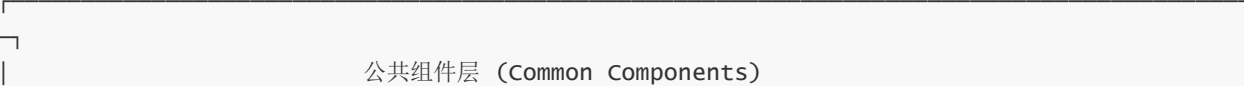
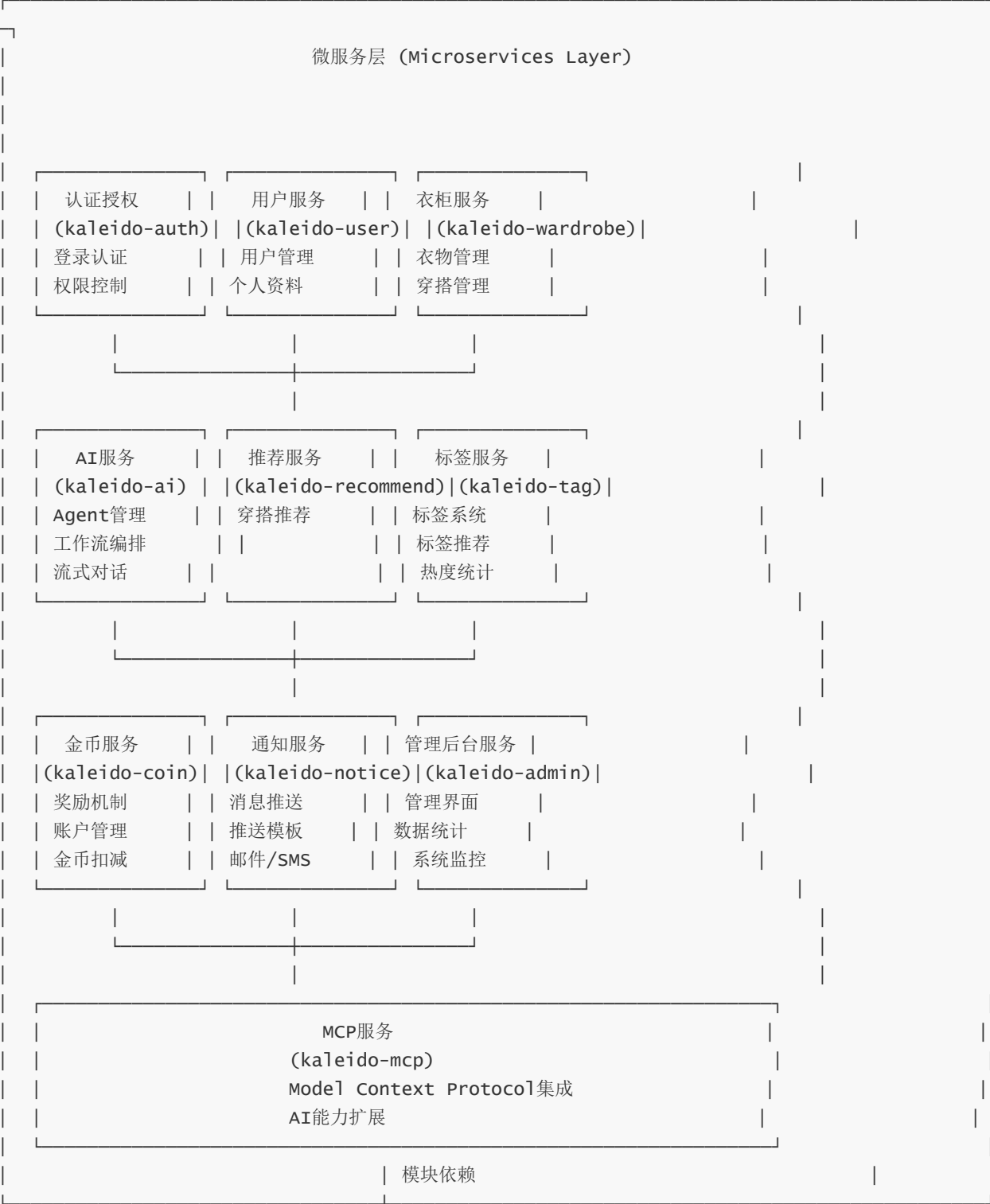
1. 外层可以依赖内层，内层不能依赖外层
2. 依赖方向：Trigger → Application → Domain ← Infrastructure
3. 领域层是核心，不依赖任何其他层

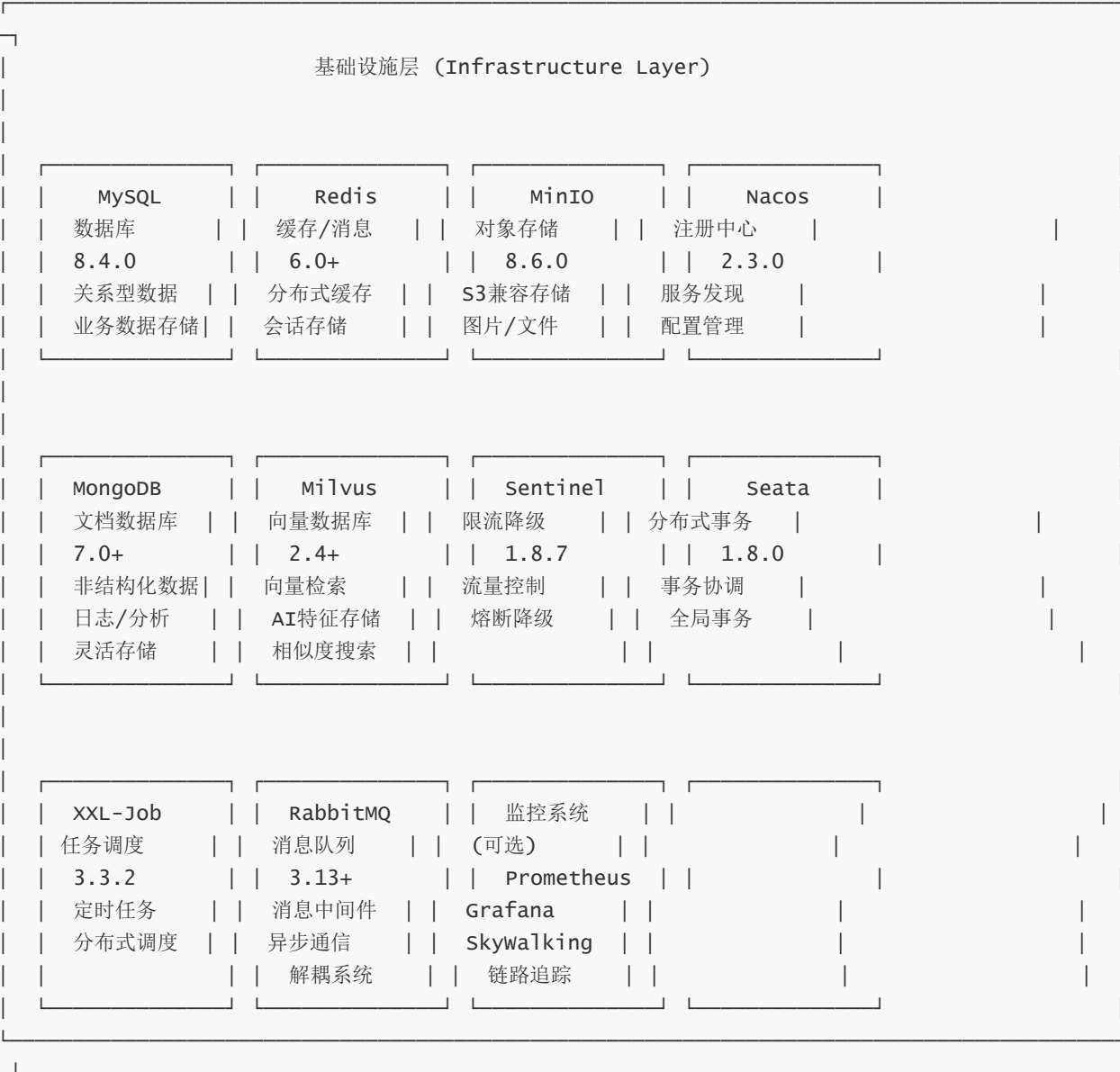
系统架构

Kaleido采用现代化的微服务架构设计，将系统拆分为多个独立的服务模块，每个模块负责特定的业务功能。这种架构设计提供了高可扩展性、高可用性和易于维护的优势。

整体架构图







后端技术

核心框架

技术	版本	说明
Java	21	开发语言
Spring Boot	3.5.6	应用框架
Spring Cloud	2025.0.0	微服务框架
Spring Cloud Alibaba	2025.0.0.0	阿里云微服务组件
Spring AI	1.1.2	AI集成框架
MyBatis-Plus	3.5.9	ORM框架

数据库

技术	版本	说明
MySQL	8.4.0	关系型数据库
MongoDB	7.0+	文档数据库，用于非结构化数据存储
Milvus	2.4+	向量数据库，用于AI特征存储和相似度搜索
MinIO	8.6.0	对象存储
shardingsphere-jdbc	5.5.2	分库分表

缓存

技术	版本	说明
Redis	6+	缓存
Redisson	3.52.0	Redis客户端
JetCache	2.7.8	多级缓存框架
Caffeine	3.2.3	本地缓存

安全

技术	版本	说明
Sa-Token	1.44.0	权限认证框架
Sentinel	1.8.7	流量控制和服务降级

分布式

技术	版本	说明
Dubbo	3.3.0	RPC框架
Nacos	2.3.0	服务注册与配置中心
Seata	1.8.0	分布式事务解决方案
XXL-Job	3.3.2	分布式任务调度

消息队列

技术	版本	说明
RabbitMQ	3.13+	消息中间件，用于异步通信和解耦系统
Spring AMQP	3.1.8	RabbitMQ集成框架

AI

技术	版本	说明
Spring AI	1.1.2	AI集成框架
Milvus	2.4+	向量数据库，用于AI特征存储和相似度搜索

开发工具

技术	版本	说明
Smart-doc	2.7.7	API文档生成
MapStruct	1.6.3	对象映射工具
Lombok	1.18.36	代码简化工具
Hutool	5.8.38	Java工具库
PageHelper	2.1.1	MyBatis分页插件

性能优化

技术	版本	说明
Dynamic-tp	1.2.2-x	动态线程池

前端技术

TODO

项目结构

kaleido-server/

```

├─ kaleido-common/                # 公共模块（23个子模块）
│   ├─ kaleido-aop/              # AOP切面
│   ├─ kaleido-api/              # API接口定义
│   ├─ kaleido-base/             # 基础工具类
│   ├─ kaleido-cache/            # 缓存模块
│   ├─ kaleido-distribute/        # 分布式相关
│   ├─ kaleido-doc/              # 文档生成
│   ├─ kaleido-ds/               # 数据源
│   ├─ kaleido-dynamic-tp/        # 动态线程池
│   ├─ kaleido-file/             # 文件服务
│   ├─ kaleido-job/              # 任务调度
│   ├─ kaleido-limiter/          # 限流器
│   ├─ kaleido-lock/             # 分布式锁
│   ├─ kaleido-mq/              # 消息队列
│   ├─ kaleido-nacos/            # 配置中心
│   ├─ kaleido-rpc/              # RPC相关
│   ├─ kaleido-sa-token/          # 权限认证
│   ├─ kaleido-seata/            # 分布式事务
│   ├─ kaleido-sentinel/         # 限流降级
│   ├─ kaleido-sms/              # 短信服务
│   └─ kaleido-web/              # web相关配置
│
├─ kaleido-gateway/              # API网关
├─ kaleido-auth/                 # 认证授权服务
├─ kaleido-biz/                  # 业务服务集合
│   ├─ kaleido-user/             # 用户服务
│   ├─ kaleido-wardrobe/         # 衣柜服务
│   ├─ kaleido-ai/               # AI服务
│   ├─ kaleido-recommend/        # 推荐服务
│   ├─ kaleido-tag/              # 标签服务
│   ├─ kaleido-coin/             # 金币服务
│   └─ kaleido-mcp/              # MCP服务
├─ kaleido-notice/               # 通知服务
├─ kaleido-admin/                # 管理后台服务
├─ doc/                          # 项目文档
├─ scripts/                      # 脚本文件
├─ .gitignore                    # Git忽略配置
├─ pom.xml                       # 父POM文件
└─ README.md                     # 项目说明文档

```

模块功能说明

1. kaleido-common 公共模块

- **kaleido-base**: 基础工具类、异常处理、通用枚举等
- **kaleido-web**: Web相关配置、统一响应、全局异常处理
- **kaleido-api**: 服务间API接口定义，使用Dubbo或OpenFeign
- **kaleido-doc**: API文档生成，集成Smart-doc
- **kaleido-rpc**: RPC调用相关配置和工具
- **kaleido-distribute**: 分布式ID生成、分布式锁等
- **kaleido-aop**: 全局AOP切面，如日志、权限、事务等
- **kaleido-sms**: 短信服务集成，支持多家短信提供商
- **kaleido-sa-token**: 权限认证框架集成
- **kaleido-job**: 分布式任务调度，集成XXL-Job
- **kaleido-dynamic-tp**: 动态线程池管理
- **kaleido-mq**: 消息队列抽象，支持多种消息中间件

- **kaleido-file**: 文件上传下载服务，支持多种存储
- **kaleido-sentinel**: 限流降级配置
- **kaleido-limiter**: 限流器实现
- **kaleido-seata**: 分布式事务配置
- **kaleido-lock**: 分布式锁实现
- **kaleido-nacos**: 配置中心客户端

2. kaleido-gateway API网关

- 基于Spring Cloud Gateway实现
- 统一入口，负责路由转发、负载均衡
- Token令牌验证和权限校验
- 请求日志记录和审计
- 跨域请求处理

3. kaleido-auth 认证授权服务

- 用户注册、登录、注销
- 多因素认证支持
- 权限角色管理
- 会话管理和单点登录
- OAuth2.0第三方登录
- 安全审计和日志

4. kaleido-biz 业务服务

- **kaleido-user**: 用户信息管理、个人资料、用户统计
- **kaleido-wardrobe**: 衣物管理、穿搭管理、穿着记录
- **kaleido-ai**: AI对话、图像识别、智能推荐
- **kaleido-recommend**: 推荐算法、个性化推荐
- **kaleido-tag**: 标签管理、标签推荐
- **kaleido-coin**: 积分系统、奖励机制
- **kaleido-mcp**: MCP协议集成，扩展AI能力

5. kaleido-notice 通知服务

- 站内消息通知
- 邮件通知
- 短信提醒
- 实时推送
- 通知模板管理

6. kaleido-admin 管理后台

- 用户管理后台
- 内容审核后台
- 数据统计后台
- 系统监控后台
- 配置管理后台

功能清单

管理后台功能清单

1. 管理员管理

- **管理员信息管理**
 - 更新管理员信息
 - 查询管理员自身信息
 - 根据ID查询管理员信息
 - 分页查询管理员列表
- **管理员状态管理**
 - 冻结管理员
 - 解冻管理员
- **角色分配管理**
 - 分配角色给管理员
- **权限查询**
 - 获取管理员的所有权限
 - 获取管理员的目录和菜单树

2. AI管理

- **Agent管理**
 - 创建Agent
 - 更新Agent信息
 - 启用/禁用Agent
 - 添加工具到Agent
 - 从Agent移除工具
 - 查询Agent详情
 - 根据编码查询Agent
 - 查询Agent列表
- **workflow管理**
 - 创建工作流
 - 更新工作流
 - 启用/禁用工作流
 - 查询工作流详情
 - 根据编码查询工作流
 - 查询工作流列表

3. 品牌管理

- **品牌操作**
 - 创建品牌
 - 更新品牌信息
 - 删除品牌（逻辑删除）
- **品牌查询**
 - 查询所有品牌列表
 - 根据ID查询品牌详情

4. 字典管理

- **字典操作**
 - 创建字典
 - 更新字典
 - 删除字典
- **字典查询**
 - 分页查询字典列表

5. 文件管理

- **文件操作**
 - 上传文件到MinIO

6. 通知管理

- **通知模板管理**
 - 添加通知模板
 - 根据模板编码获取模板详情
 - 分页查询通知模板列表
- **通知管理**
 - 根据ID获取通知详情
 - 根据目标地址获取通知列表
 - 分页查询通知列表

7. 权限管理

- **权限操作**
 - 创建权限
 - 更新权限信息
 - 更新权限编码
 - 删除权限
- **权限查询**
 - 根据ID查询权限信息
 - 根据编码查询权限信息
 - 根据父权限ID查询子权限列表
 - 获取权限树

8. 角色管理

- **角色操作**
 - 创建角色
 - 更新角色信息
 - 删除角色
 - 分配权限给角色
- **角色查询**
 - 根据ID查询角色信息
 - 根据编码查询角色信息
 - 获取角色列表

9. 用户管理

- 用户操作
 - 冻结用户
 - 解冻用户
 - 删除用户（软删除）
- 用户查询
 - 根据用户ID查询用户信息
 - 根据手机号查询用户信息
 - 分页查询用户列表

用户服务功能清单

1. 用户信息管理

- 用户信息查询
 - 查询用户自身信息
 - 分页查询用户列表
- 用户信息更新
 - 修改用户昵称
 - 更新用户头像

授权服务功能清单

1. 用户认证管理

- 用户注册登录
 - 用户注册
 - 用户登录

2. 短信验证码管理

- 短信验证码
 - 发送短信验证码

衣柜服务功能清单

1. 品牌管理

- 品牌查询
 - 查询所有品牌列表
 - 根据ID查询品牌详情

2. 服装管理

- 服装操作
 - 创建服装（包含图片）
 - 更新服装信息（包含图片）
 - 删除服装
- 服装查询
 - 查询用户服装列表
 - 根据ID查询服装详情

- **服装标签管理**
 - 为服装添加标签
 - 从服装移除标签

3. 位置管理

- **位置操作**
 - 创建位置（包含图片）
 - 更新位置信息
 - 删除位置（逻辑删除）
- **位置查询**
 - 查询用户位置列表
 - 根据ID查询位置详情
- **位置标签管理**
 - 为位置添加标签
 - 从位置移除标签

4. 穿搭管理

- **穿搭操作**
 - 创建穿搭（包含服装和图片）
 - 更新穿搭信息（包含服装和图片）
 - 删除穿搭
- **穿搭查询**
 - 根据用户ID查询穿搭列表
 - 根据ID查询穿搭详情
- **穿着记录管理**
 - 记录穿搭穿着
 - 查询穿搭穿着记录

标签服务功能清单

1. 标签管理

- **标签操作**
 - 创建标签
 - 更新标签
- **标签查询**
 - 根据ID查询标签
 - 查询标签列表（按标签类型编码）

AI服务功能清单

1. Agent管理

- **Agent聊天**
 - 与Agent进行聊天
 - 检查Agent是否可用
 - 获取Agent状态
- **Agent查询**
 - 查询Agent详情

- 根据编码查询Agent
- 查询Agent列表

2. 聊天管理

- AI聊天
 - 与AI进行聊天（搭配推荐）
 - 保存衣物信息到向量数据库

3. 会话管理

- 会话操作
 - 创建会话
 - 更新会话标题
 - 删除会话
- 会话查询
 - 查询会话详情
 - 查询用户会话列表
 - 查询用户活跃会话列表

4. workflows管理

- 工作流查询
 - 查询工作流详情
 - 根据编码查询工作流
 - 查询启用的工作流列表

推荐服务功能清单

1. 推荐记录管理

- 推荐记录操作
 - 创建AI穿搭推荐
 - 查询单个推荐记录详情
 - 查询用户的推荐记录列表

API 文档

Kaleido 项目使用 **Smart-doc** 作为 API 文档生成工具，通过解析 Java 注释自动生成高质量的 API 文档。

生成文档的命令

每个服务模块都配置了 Smart-doc Maven 插件，可以通过以下命令生成 API 文档：

```
# 在项目根目录生成所有服务的文档
mvn compile smart-doc:html

# 或在单个服务目录下生成该服务的文档
cd kaleido-auth
mvn compile smart-doc:html
```

访问生成的文档

文档生成后，可以在项目根目录的 `doc/api/` 目录下找到各服务的 API 文档：

- **认证服务**： `doc/api/auth/index.html`
- **用户服务**： `doc/api/user/index.html`
- **衣柜服务**： `doc/api/wardrobe/index.html`
- **AI服务**： `doc/api/ai/index.html`
- **标签服务**： `doc/api/tag/index.html`
- **推荐服务**： `doc/api/recommend/index.html`
- **金币服务**： `doc/api/coin/index.html`
- **通知服务**： `doc/api/notice/index.html`
- **管理后台**： `doc/api/admin/index.html`

注意：文档生成路径在各自的 `smart-doc-config.json` 文件中配置，默认输出到项目根目录的 `doc/api/{服务名}/` 目录下。

为什么选择 Smart-doc 而不是 Swagger

1. 基于注释生成，代码侵入性更小：
2. 反向推动开发者编写注释，提升代码质量：
3. 生成静态文档，性能更好：
4. 更灵活的文档定制：
5. 其实项目最初使用的是 Swagger，但本着我的项目我做主的原则，全面替换成了 Smart-doc：

后续计划

1. **集成ELK可视化日志管理**
 - 部署Elasticsearch、Logstash、Kibana (ELK) 技术栈
 - 实现分布式日志收集、存储和可视化分析
 - 提供实时日志查询、错误追踪和性能监控能力
2. **Grafana + Prometheus监控体系**
 - 搭建Prometheus时序数据库，收集系统指标数据
 - 配置Grafana可视化仪表盘，展示系统运行状态
 - 实现服务健康检查、资源使用率监控和告警机制
3. **SkyWalking链路追踪**
 - 集成Apache SkyWalking分布式追踪系统
 - 实现微服务调用链路的可视化追踪
 - 分析服务间调用性能，定位瓶颈和异常
4. **前端页面开发**
 - 开发用户友好的Web管理界面
 - 实现响应式设计，支持PC和移动端访问
 - 构建直观的数据可视化图表和操作界面