宠物综合管理平台系统手册

1. 概述

本手册旨在为宠物综合管理平台的部署、运行和维护提供详细指导。平台采用前后端分离架构，后端基于Spring Boot，前端基于Vue 3，并支持网页端和手机小程序端访问（小程序端需额外开发）。核心功能围绕NFC扫码识别宠物身份展开，涵盖用户端、商家端和平台管理端。

2. 技术栈

2.1 后端技术栈

技术	版本	描述
Java	17	编程语言
Spring Boot	3.2.x	核心框架，用于快速构建RESTful API
Spring Security	6.2.x	认证与授权框架，提供JWT支持
Spring Data JPA	3.2.x	持久层框架，简化数据库操作
Hibernate	6.4.x	JPA实现，对象关系映射
Maven	3.8.x	项目管理和构建工具
H2 Database	2.2.x	开发环境默认内存数据库，可替换为MySQL/PostgreSQL
JWT	0.11.5	JSON Web Token，用于无状态认证

2.2 前端技术栈

技术	版本	描述
Vue	3.4.x	渐进式JavaScript框架，用于构建用户界面
TypeScript	5.3.x	JavaScript超集，提供类型安全
Vite	5.0.x	前端构建工具，提供极速开发体验
Pinia	2.1.x	Vue官方推荐的状态管理库
Vue Router	4.2.x	Vue官方路由管理器
Element Plus	2.5.x	基于Vue 3的企业级UI组件库，提供ERP风格组件
Axios	1.6.x	基于Promise的HTTP客户端，用于前后端通信
pnpm	10.18.x	包管理工具，高效且节省磁盘空间

3. 源码获取与项目结构

源码将以压缩包形式提供。解压后，项目结构如下：

Plain Text

```
pet-management-platform/
├── backend/                    # Spring Boot后端项目
│   ├── pom.xml                 # Maven项目配置文件
│   └── src/                    # 源代码目录
│       ├── main/
│       │   ├── java/           # Java源代码
│       │   │   └── com/petmanagement/petmanagementbackend/
│       │   │       ├── config/          # 配置类，如DataLoader
│       │   │       ├── controllers/     # RESTful API控制器
│       │   │       ├── models/          # 实体类/数据模型
│       │   │       ├── payload/         # 请求/响应DTO
│       │   │       ├── repository/      # JPA数据仓库接口
│       │   │       └── security/        # Spring Security相关配置和工具
│       │   └── resources/      # 资源文件，如application.properties
│       └── test/               # 测试代码
├── frontend/                   # Vue 3前端项目
│   ├── package.json            # pnpm项目配置文件
│   ├── vite.config.ts          # Vite配置文件
│   ├── tsconfig.json           # TypeScript配置文件
│   ├── index.html              # 入口HTML文件
│   └── src/                    # 源代码目录
```

```
│       ├── api/                  # API请求封装
│       ├── assets/               # 静态资源
│       ├── components/           # 可复用组件
│       ├── router/               # Vue Router配置
│       ├── stores/               # Pinia状态管理模块
│       ├── types/                # TypeScript类型定义
│       ├── utils/                # 工具函数，如request封装
│       ├── views/                # 页面组件
│       │   ├── Login.vue         # 登录页面
│       │   ├── Layout.vue        # 整体布局页面
│       │   ├── Dashboard.vue     # 仪表盘页面
│       │   ├── pets/             # 宠物管理页面
│       │   │   └── PetList.vue   # 宠物列表及CRUD
│       │   ├── nfc/              # NFC吊牌管理页面
│       │   │   └── NfcTagList.vue # NFC吊牌列表及CRUD、绑定/解绑
│       │   ├── services/         # 服务管理页面
│       │   │   └── ServiceList.vue # 服务列表及CRUD
│       │   ├── orders/           # 订单管理页面
│       │   │   └── OrderList.vue # 订单列表及CRUD、NFC扫码处理
│       │   ├── base-data/        # 基础数据管理页面
│       │   │   └── BaseDataList.vue # 基础数据列表及CRUD
│       │   └── health/           # 健康记录管理页面
│       │       └── HealthRecordList.vue # 健康记录列表及CRUD
│       └── App.vue               # 根组件
│       └── main.ts               # Vue应用入口文件
└── system_manual.md              # 本系统手册
```

4. 后端部署指南

4.1 环境准备

1. **安装Java Development Kit (JDK) 17**：
   - 访问 Oracle官网 或 AdoptOpenJDK 下载适用于您Windows系统的JDK 17安装包（通常是 `.msi` 或 `.zip`）。
   - 按照安装向导进行安装。
   - **配置环境变量**：
     - 设置 `JAVA_HOME` 环境变量指向JDK安装路径（例如：`C:\Program Files\Java\jdk-17`）。
     - 将 `%JAVA_HOME%\bin` 添加到系统的 `Path` 环境变量中。
   - 验证安装：打开命令提示符（CMD）或PowerShell，运行 `java -version` 和 `javac -version`。
2. **安装Maven 3.8.x**：

- 访问 Apache Maven官网 下载Maven 3.8.x的二进制文件（通常是 `.zip` ）。
- 解压到您选择的目录（例如： `C:\apache-maven-3.8.x` ）。
- **配置环境变量**：
  - 设置 `M2_HOME` 环境变量指向Maven安装路径（例如： `C:\apache-maven-3.8.x` ）。
  - 将 `%M2_HOME%\bin` 添加到系统的 `Path` 环境变量中。
- 验证安装：打开命令提示符（CMD）或PowerShell，运行 `mvn -v` 。

3. **数据库**：
   - **开发环境（默认H2内存数据库）**：无需额外配置，应用启动时会自动创建数据库。数据会在应用停止后丢失。
   - **生产环境（推荐MySQL或PostgreSQL）**：
     - 安装并配置您选择的数据库服务（例如 MySQL Community Server 或 PostgreSQL）。
     - 创建数据库和用户。
     - **重要**：修改 `backend\pet-management-backend\src\main\resources\application.properties` 文件中的数据库连接配置。

4.2 配置JWT密钥

打开 `backend\pet-management-backend\src\main\resources\application.properties` 文件，修改 `app.jwt.secret` 为一个足够长且安全的随机字符串。例如：

```
Plain Text
app.jwt.secret=YourSuperSecretJwtKeyThatIsAtLeast256BitsLongAndRandomlyGenerated
```

4.3 编译与运行

1. 打开命令提示符（CMD）或PowerShell，进入后端项目根目录：
2. 编译项目：
3. 运行项目：

后端服务默认运行在 `http://localhost:8080` 。

5. 前端部署指南

5.1 环境准备

1. **安装Node.js (推荐LTS版本，如18.x或20.x)**：
   - 访问 Node.js官网 下载适用于您Windows系统的安装包（通常是 `.msi` ）。
   - 按照安装向导进行安装，确保勾选“Add to PATH”选项。
   - 验证安装：打开命令提示符（CMD）或PowerShell，运行 `node -v` 和 `npm -v` 。
2. **安装pnpm 10.18.x**：
   - 如果未安装，可以使用npm安装：
   - 验证安装： `pnpm -v`

5.2 配置API代理

前端项目已配置Vite代理，将 `/api` 请求转发到后端服务。如果后端服务不在 `http://localhost:8080` ，请修改 `frontend\vite.config.ts` 文件中的 `target` ：

TypeScript

```
// vite.config.ts
export default defineConfig({
  // ...
  server: {
    port: 5173,
    proxy: {
      '/api': {
        target: 'http://localhost:8080', // 修改为您的后端服务地址
        changeOrigin: true
      }
    }
  }
})
```

5.3 编译与运行

1. 打开命令提示符（CMD）或PowerShell，进入前端项目根目录：
2. 安装依赖：
3. 运行开发服务器：
4. 构建生产版本：

6. 初始用户与角色

系统启动时， `DataLoader` 会初始化 `ROLE_PET_OWNER` 、 `ROLE_BUSINESS` 和 `ROLE_ADMIN` 三个角色。您可以通过注册接口创建用户，并手动在数据库中修改其角色，或者通过管理员接口进行管

理（待开发）。

7. 后续开发建议

- **手机小程序端**：基于后端API，开发微信小程序或支付宝小程序等移动端应用。
- **NFC扫码适配**：研究不同移动平台（iOS/Android）的NFC API，实现手机小程序端的NFC扫码功能。
- **数据统计报表**：根据需求开发数据统计和可视化功能。
- **文件上传**：集成文件存储服务（如MinIO、阿里云OSS）实现头像、健康记录附件等文件的上传。
- **消息通知**：集成消息队列（如RabbitMQ）和消息推送服务（如短信、邮件）实现日程提醒等功能。

8. 常见问题与故障排除

- **后端启动失败**：
  - 检查 `application.properties` 中的数据库配置是否正确。
  - 检查端口 `8080` 是否被占用。
  - 查看控制台日志，根据错误信息进行排查。
- **前端页面空白或API请求失败**：
  - 确保后端服务已成功启动。
  - 检查 `vite.config.ts` 中的API代理配置是否正确。
  - 打开浏览器开发者工具，查看网络请求和控制台错误。
- **权限问题**：
  - 确保登录用户具有访问对应接口的权限。
  - 检查JWT令牌是否正确生成和传递。

Manus AI 撰写 **日期**：2025年10月14日