

# 系统架构

---

## 整体架构

```
space-management/
├── src/
│   ├── components/    # React组件
│   ├── store/         # Redux状态管理
│   ├── types/         # TypeScript类型定义
│   ├── utils/         # 工具函数
│   ├── services/      # API服务
│   └── assets/        # 静态资源
```

## 核心模块

### 1. Canvas模块

- `components/Canvas/index.tsx`: 画布容器组件，负责2D/3D视图切换
- `components/Canvas/Canvas2D.tsx`: 基于Canvas 2D的平面绘图引擎
- `components/Canvas/Canvas3D.tsx`: Three.js驱动的3D渲染组件
- `components/Canvas/WallDrawing.tsx`: 墙体绘制专用组件
- `components/Canvas/Element3D.tsx`: 3D图元实例化渲染组件

### 2. 状态管理

- `store/spaceSlice.ts`: 空间数据状态管理，包含图元、视图模式等
- `store/commandSlice.ts`: 命令系统，实现撤销/重做功能
- `store/fileSlice.ts`: 文件操作状态管理
- `store/authSlice.ts`: 用户认证状态管理

### 3. 面板组件

- `ElementPanel/index.tsx`: 元素选择面板，提供图元库
- `PropertyPanel/index.tsx`: 属性编辑面板，编辑选中图元属性
- `FileDialog/index.tsx`: 文件操作对话框
- `Header/index.tsx`: 顶部工具栏，包含文件操作和视图切换

### 4. 工具类

- `utils/elementUtils.ts`: 图元操作工具函数
- `utils/geometryUtils.ts`: 几何计算工具函数
- `utils/commandUtils.ts`: 命令系统工具函数
- `utils/selectionUtils.ts`: 选择操作工具函数

## 数据流

---

## 技术实现

### 核心功能实现

- 基于Ant Design ProComponents的布局系统
- 使用React Router v6实现嵌套路由和动态路由
- 集成Redux Toolkit进行类型安全的状态管理
- 实现基于JWT的OAuth2认证流程（登录/注册/刷新令牌）
- 采用原子化组件设计架构
- 实现命令模式(Command Pattern)的撤销/重做系统
- 使用Zod进行表单验证和类型校验

### 状态管理

- 使用Redux Toolkit管理全局状态（包含spaceSlice、commandSlice、fileSlice、authSlice）
- 实现基于命令模式(Command Pattern)的撤销/重做系统（最大支持100步历史记录）
- 支持本地IndexedDB存储和REST API云端同步
- 实现基于JWT Token的用户认证系统（包含自动刷新令牌机制）
- 使用Redux Persist进行状态持久化（白名单配置：auth、file）
- 通过Redux Thunk中间件处理异步操作
- 使用TypeScript类型系统保障状态安全
- 状态结构版本化管理（通过migration配置）

### 图元系统

- 支持多种图元类型：墙体、座位、门窗、会议室等
- 每种图元都有独立的属性和行为
- 实现了图元的样式系统，支持默认、悬停、选中等状态
- 支持图元的组合和关联