

这是一个非常关键的步骤。为了确保你在新设备上能够顺利运行这个项目，你需要导出的不仅仅是代码文件，还包括项目结构配置和静态资源。

以下是**必须导出（保留）的核心文件和文件夹清单，以及不需要导出的文件**：

## 📁 1. 核心源码文件夹（必须导出）

这是我们所有逻辑所在的地方，是最重要的部分。

- **apps/client/src/** (整个文件夹)
  - **components/**: 包含了我们辛苦优化的组件。
    - FloorPlanEditor.tsx (核心编辑器，包含高性能渲染逻辑)
    - DeviceSidebar.tsx (侧边栏，包含树状图和 Missing 列表)
    - SiteManager.tsx (楼层管理弹窗)
  - **store/**: 包含了状态管理。
    - useSiteStore.ts (地图、坐标、楼层数据)
    - useTopologyStore.ts (拓扑结构、设备状态)
  - **utils/**: 工具函数。
    - fileParser.ts (强大的 HTML/JSON 解析器)
    - topologyTree.ts (树结构生成算法)
    - storage.ts (IndexedDB 图片存储逻辑)
  - App.tsx (主入口)
  - main.tsx / index.tsx
  - index.css (包含 Tailwind 引用)
- **apps/server/src/** (整个文件夹)
  - 后端 NestJS 代码（如果你的后端逻辑也在这里）。

## 📁 2. 静态资源（必须导出）

如果缺少这些，图标会显示不出来，或者无法进行测试。

- **apps/client/public/** (整个文件夹)
  - 特别是 **apps/client/public/assets/icons/**: 这里面存放了 Leader.svg,

Router.svg 以及 smoke.svg 等所有设备图标。

- **测试数据文件：**
  - 例如你上传的 Topology-Tool-148-result.html，建议备份一份以便在新电脑上测试导入功能。

### 3. 配置文件 (必须导出)

新环境安装依赖和运行项目全靠这些文件。

- **根目录及子目录下的 package.json：**
  - 根目录的 package.json
  - apps/client/package.json
  - apps/server/package.json
- **锁文件：**
  - pnpm-lock.yaml (或者 package-lock.json / yarn.lock)：确保新电脑安装的版本与现在一致。
- **构建配置：**
  - apps/client/vite.config.ts
  - apps/client/tailwind.config.js
  - apps/client/postcss.config.js
  - apps/client/tsconfig.json
  - tsconfig.base.json (如果有)

### 4. 不需要导出的文件 (减小体积)

以下文件夹**不要**复制，因为它们体积巨大且可以自动生成：

- **node\_modules/**：这是最占空间的文件，在新电脑上运行 npm install 或 pnpm install 即可重新下载。
  - **dist/** 或 **build/**：这是编译后的产物，不需要保留。
  - **.vite/**：Vite 的缓存文件夹，不需要。
-

## 迁移步骤指南

### 1. 打包：

将除了 `node_modules`, `dist`, `.vite` 以外的所有文件压缩成一个 `.zip` 包。

或者，如果你使用了 *Git*，直接将代码推送到 *GitHub/GitLab* 仓库。

### 2. 新设备环境准备：

- 安装 **Node.js** (建议 v18 或 v20 LTS)。
- 安装包管理器 (如果用的是 pnpm)： `npm install -g pnpm`。

### 3. 恢复项目：

- 解压文件。
- 在根目录打开终端。
- 运行安装命令： `pnpm install` (或 `npm install`)。

### 4. 启动：

- `pnpm dev` (前端)
- `pnpm start:dev` (后端，如果有)

## 最终检查清单 (Checklist)

在打包前，请最后确认一下 `apps/client/src/components/FloorPlanEditor.tsx` 是否包含了我们刚才修复的 **V24** 版本代码（即包含 `SingleDeviceNode` 且 `Nodes` 组件有 `key={...}` 强制刷新逻辑）。如果不确定，可以再运行一次刚才的 V24 修复脚本。