

# 双轨人生项目协作开发指南

## 项目概述

双轨人生项目已成功从GitHub仓库 <https://github.com/toybot1981/dual-track-life.git> 下载并部署。本指南将帮助开发团队进行高效的协作开发。

## 项目架构

### 技术栈

- 前端: Vue 3 + TypeScript + Vite + Tailwind CSS
- 后端: Spring Boot (Java) / FastAPI (Python)
- 数据库: SQLite / MySQL
- 状态管理: Pinia
- 路由: Vue Router
- 国际化: Vue I18n

### 目录结构

```
dual-track-life-main/  
├── frontend/           # Vue.js 前端应用  
├── backend-java/       # Spring Boot 后端  
├── database/           # 数据库初始化文件  
├── docs/               # 项目文档  
└── package.json        # 项目配置
```

## 开发环境设置

### 1. 克隆项目

```
# 下载项目ZIP文件  
wget https://github.com/toybot1981/dual-track-life/archive/refs/  
heads/main.zip  
unzip main.zip  
cd dual-track-life-main
```

## 2. 前端开发环境

```
cd frontend
npm install
npm run dev
```

开发服务器: <http://localhost:5173>

## 3. 构建和部署

```
npm run build
# 构建文件位于 dist/ 目录
```

## 开发工作流

### Git 工作流建议

1. **主分支**: `main` - 稳定版本
2. **开发分支**: `develop` - 开发版本
3. **功能分支**: `feature/功能名` - 新功能开发
4. **修复分支**: `hotfix/问题描述` - 紧急修复

### 代码提交规范

```
feat: 新功能
fix: 修复bug
docs: 文档更新
style: 代码格式调整
refactor: 代码重构
test: 测试相关
chore: 构建过程或辅助工具的变动
```

## 核心功能模块

### 1. 用户认证系统

- **文件位置**: `src/stores/auth.ts`, `src/views/LoginView.vue`
- **功能**: 用户登录、注册、JWT认证
- **API**: `/auth/login`, `/auth/register`

## 2. 事件管理系统

- 文件位置: `src/stores/events.ts`, `src/views/DashboardView.vue`
- 功能: 事件CRUD、时间轴展示
- API: `/events/*`

## 3. 平行宇宙系统

- 文件位置: `src/stores/parallelUniverse.ts`, `src/components/ParallelUniverseModal.vue`
- 功能: 平行宇宙创建、管理、对比
- 特色: AI规划模式、用户自定义模式

## 4. 虚拟人生模拟

- 文件位置: `src/stores/virtualLife.ts`, `src/components/SimulationModal.vue`
- 功能: AI驱动的人生模拟
- 特色: 多种模拟场景、决策系统

## 当前部署状态

### 部署信息

- 部署地址: <https://znjotvty.manus.space>
- 构建状态:  成功 (331.35 kB)
- 构建时间: 11.54s
- 依赖安装:  完成 (421 packages)

### 项目特点

- 完整的Vue 3 + TypeScript项目结构
- 国际化支持 (中英文)
- 响应式设计
- 模块化组件架构
- 完善的状态管理

# 代码质量标准

## TypeScript 配置

- 严格模式启用
- 类型检查完整
- 接口定义规范

## Vue 3 最佳实践

- Composition API 优先
- 响应式数据管理
- 组件化开发
- Props 类型定义

## 样式规范

- Tailwind CSS 实用类
- 响应式设计
- 一致的设计系统
- 深色模式支持

## 开发任务建议

### 优先级1: 核心功能完善

1. 平行宇宙功能优化
2. 完善AI规划算法
3. 增加更多主题选择
4. 优化用户体验
5. 数据可视化增强
6. 添加图表组件
7. 统计数据展示
8. 趋势分析功能

### 优先级2: 用户体验提升

1. 界面优化
2. 动画效果增强

3. 交互反馈改进

4. 移动端适配

5. **性能优化**

6. 代码分割

7. 懒加载实现

8. 缓存策略

## 优先级3: 功能扩展

1. **社交功能**

2. 用户分享

3. 社区互动

4. 经验交流

5. **AI功能增强**

6. 智能推荐

7. 个性化建议

8. 预测分析

## 开发工具推荐

### IDE 配置

- **VS Code** + Vue Language Features (Volar)
- **WebStorm** + Vue.js 插件

### 必备扩展

- Vue Language Features (Volar)
- TypeScript Vue Plugin (Volar)
- Tailwind CSS IntelliSense
- ESLint
- Prettier

### 调试工具

- Vue DevTools
- Chrome DevTools

- Network 面板监控



## 性能监控

### 构建分析

- Bundle 大小: 331.35 kB (gzipped: 111.61 kB)
- CSS 大小: 53.05 kB (gzipped: 8.47 kB)
- 构建时间: ~11秒

### 优化建议

- 代码分割优化
- 图片资源压缩
- CDN 资源使用
- 缓存策略实施



## 安全考虑

### 前端安全

- XSS 防护
- CSRF 保护
- 敏感数据处理
- 安全的路由守卫

### API 安全

- JWT 令牌管理
- 请求验证
- 错误处理
- 数据验证



## 文档维护

### 代码文档

- JSDoc 注释
- 组件文档
- API 文档

- 类型定义

## 用户文档

- 使用指南
- 功能说明
- 故障排除
- 更新日志



## 团队协作

### 沟通渠道

- GitHub Issues - 问题跟踪
- Pull Requests - 代码审查
- 项目看板 - 任务管理

### 代码审查

- 功能完整性检查
- 代码质量评估
- 性能影响分析
- 安全性审查



## 项目路线图

### 短期目标 (1-2周)

- ☐ 平行宇宙功能完善
- ☐ 用户体验优化
- ☐ 性能调优

### 中期目标 (1-2月)

- ☐ 后端API完善
- ☐ 数据库优化
- ☐ 移动端适配

### 长期目标 (3-6月)

- ☐ AI功能增强

- ☐ 社交功能开发
- ☐ 商业化准备

---

项目已准备就绪，期待团队的精彩贡献! 🚀