

项目结构与初始配置

项目建议结构

├── android
├── assets // 静态资源
├── index.js // 入口文件
├── ios
├── patches // 有时要给第三方打补丁(patch-package)
├── scripts // 放一些自动化辅助脚本
├── src
├── components // 全局复用的非全屏组件
├── routers // 页面路由结构声明(stack, tab, drawer)
├── screens // 全屏页面组件(可以继续嵌套 components)
├── reduxState // redux 相关
├── actions.ts
├── reducers
├── sagas
├── store.ts
├── selectors.ts
├── types // 类型声明
├── utils // 辅助函数、常量、配置等
├── navService.ts
├── constants.ts
├── config.ts
├── request.ts
└── App.tsx

初始配置

原生配置

- 允许 http(ios 9, android 9 开始, 默认只允许 https)
- manifest 权限检查
 - runtime permission (android 6)
- react-native-gesture-handler // 手势组件 (react-navigation 必需)
- react-native-screens // (react-navigation 可选) 提升 react-navigation 性能的可选组件
- 删除不用的 tv 和 test 项目

js 代码配置

- 屏蔽一些无意义的警告
- 运行时权限检查
- 二级目录配置 package.json // 方便相对路径引用
- Redux 全家桶 (要点回顾) :
 - redux
 - react-redux // 链接 react 组件
 - redux-saga // 处理异步副作用
 - redux-persist // 持久记录 redux 状态
 - redux-type-actions // 增强类型提示
 - immer // 简化不可变类型状态的修改
 - reselect // 缓存 selector 以减少不必要的状态计算
 - @types/react-redux // react-redux 的类型声明

如何修复第三方的问题

- 查看是否有更新的测试版或者未发布版本

- 以 react-native-swiper 为例
- fork
 - 自己修改后发布，或者直接安装 github 地址
 - 可向原作者贡献补丁
- patch-package
 - 如果是修改原生第三方，先生成补丁，再编译
- 用yarn resolutions修复间接依赖

