

极客园移动端

项目开发环境创建



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

基于Vite创建开发环境

Vite是一个框架无关的前端工具链，可以快速的生成一个 React + TS 的开发环境，并且可以提供快速的开发体验

```
npm create vite@latest react-jike-mobile -- --template react-ts
```

说明：

1. npm create vite@latest 固定写法 （使用最新版本vite初始化项目）
2. react-ts-pro 项目名称 （可以自定义）
3. -- --template react-ts 指定项目模版位react+ts

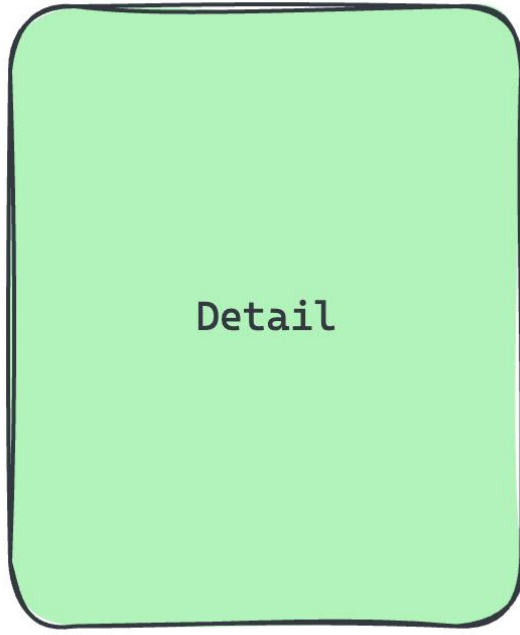
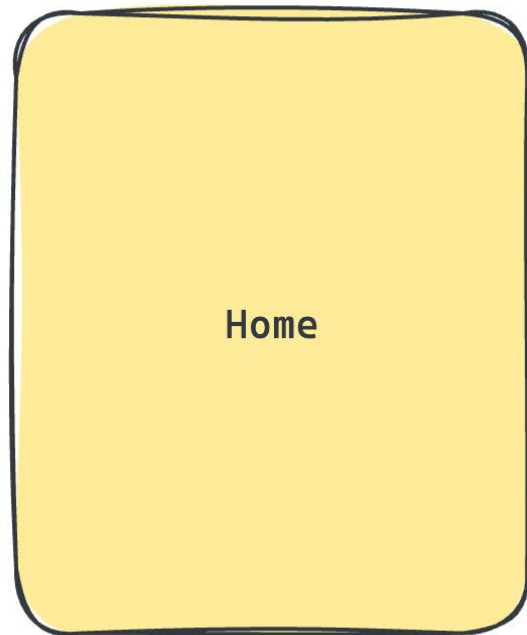
安装Ant Design Mobile

Ant Design Mobile 是 Ant Design 家族里专门针对于移动端的组件库

看文档!

初始化路由

React的路由初始化，我们采用 `react-router-dom` 进行配置



配置路径别名

场景：项目中各个模块之间的互相导入导出，可以通过@别名路径做路径简化，经过配置@相当于src目录，比如：

```
1 import Detail from '../pages/Detail'
2
3 import Detail from '@pages/Detail'
```

步骤：

1. 让Vite做路径解析（真实的路径转换）
2. 让VSCode做智能路径提示（开发者体验）

安装axios

场景：axios作为最流行的请求插件，同样是类型友好的，基于axios做一些基础的封装

1. 安装axios到项目
2. 在utils中封装http模块，主要包括接口基地址、超时时间、拦截器
3. 在utils中做统一导出

封装API模块-axios和ts的配合使用

场景：axios提供了request泛型方法，方便我们传入类型参数推导出接口返回值的类型

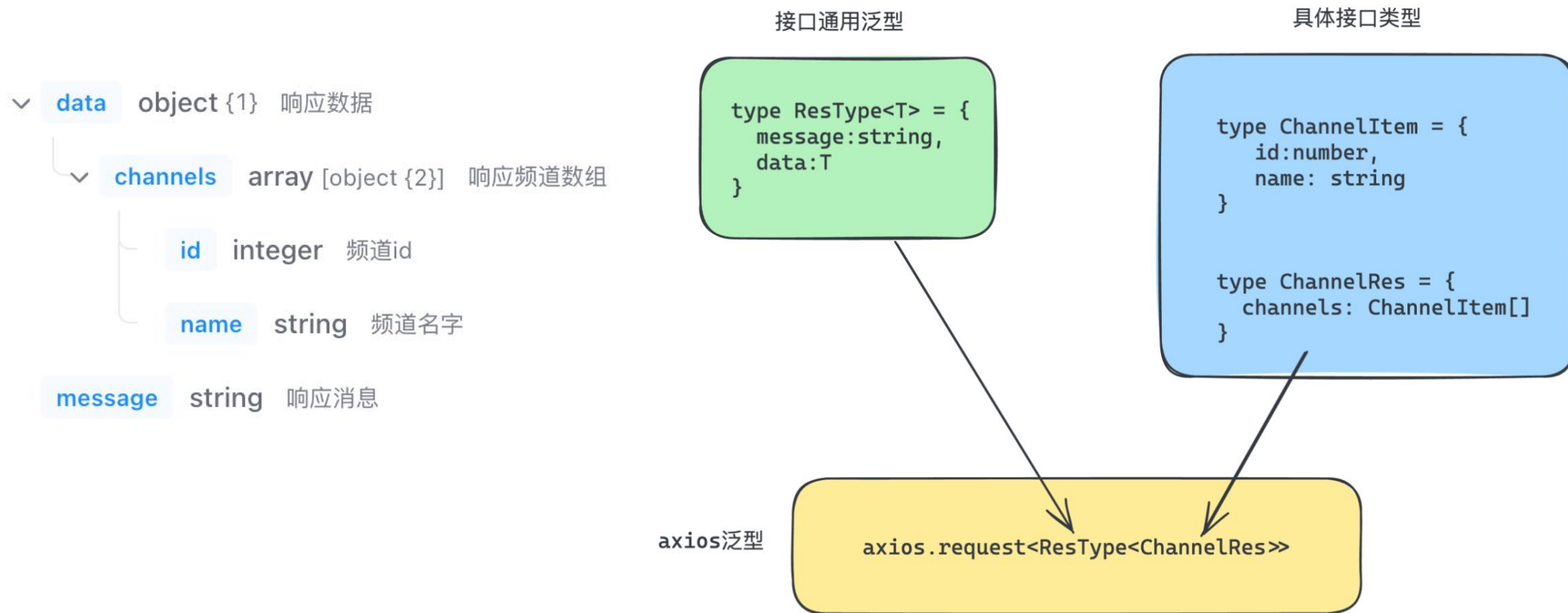
```
1 axios.request<Type>(requestConfig).then(res=>{  
2   // res.data 的类型为 Type  
3   console.log(res.data)  
4 })
```

说明：泛型参数 `Type` 的类型决定了 `res.data` 的类型

步骤：

1. 根据接口文档创建一个通用的泛型接口类型（多个接口返回值的结构是相似的）
2. 根据接口文档创建特有的接口数据类型（每个接口有自己特殊的数据格式）
3. 组合1和2的类型，得到最终传给request泛型的参数类型

封装API模块-axios和ts的配合使用-流程图



极客园移动端

Home模块开发



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

Home模块-整体组件嵌套设计

推荐 html 开发者资讯 c++ css 数



如何学习前端



啊

大大大大大

测试用例



波奇酱

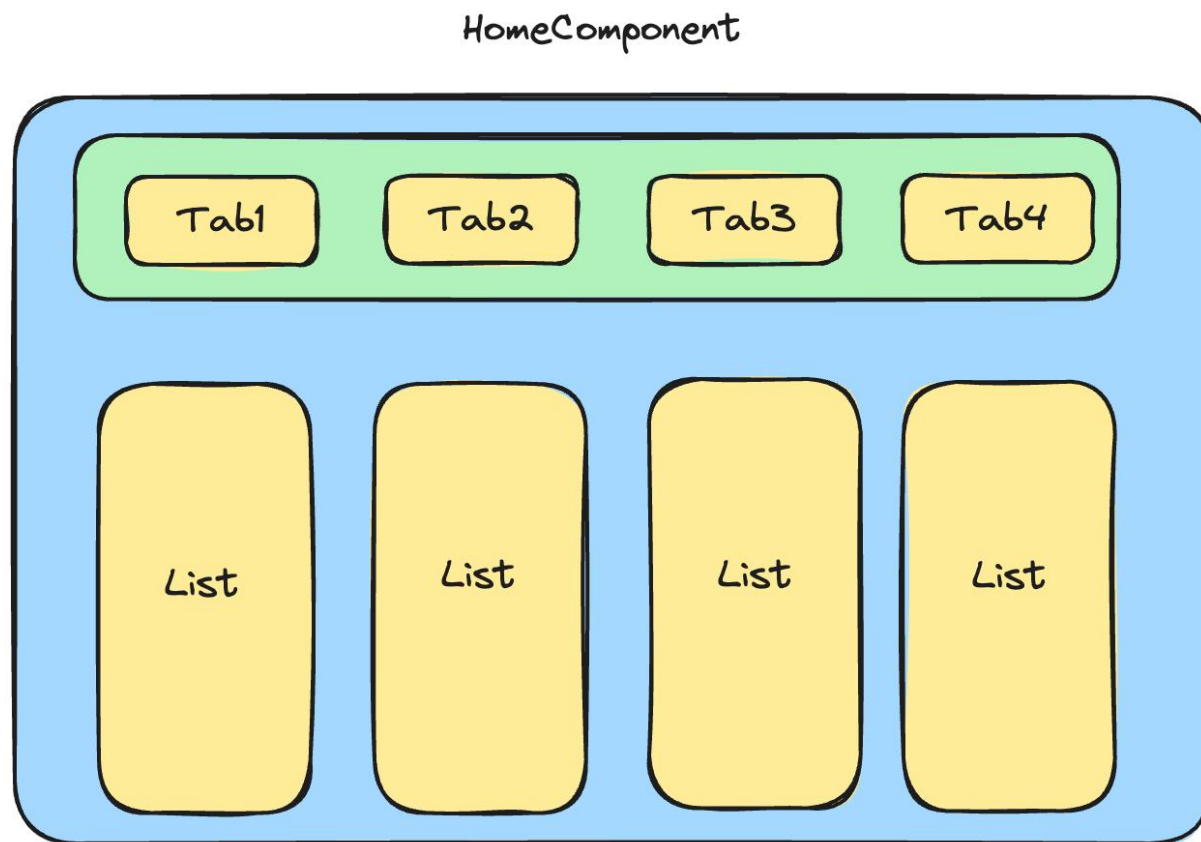
这是一条测试文章

11

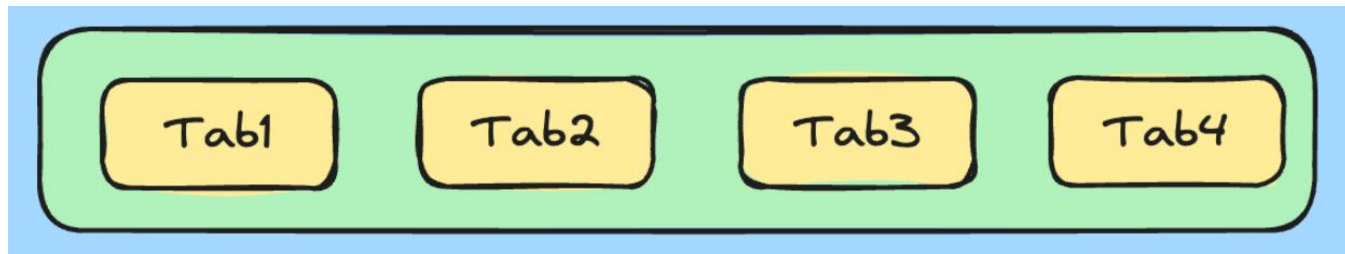


22222222333

我是一个标题



Home模块-Tabs区域实现



实现步骤:

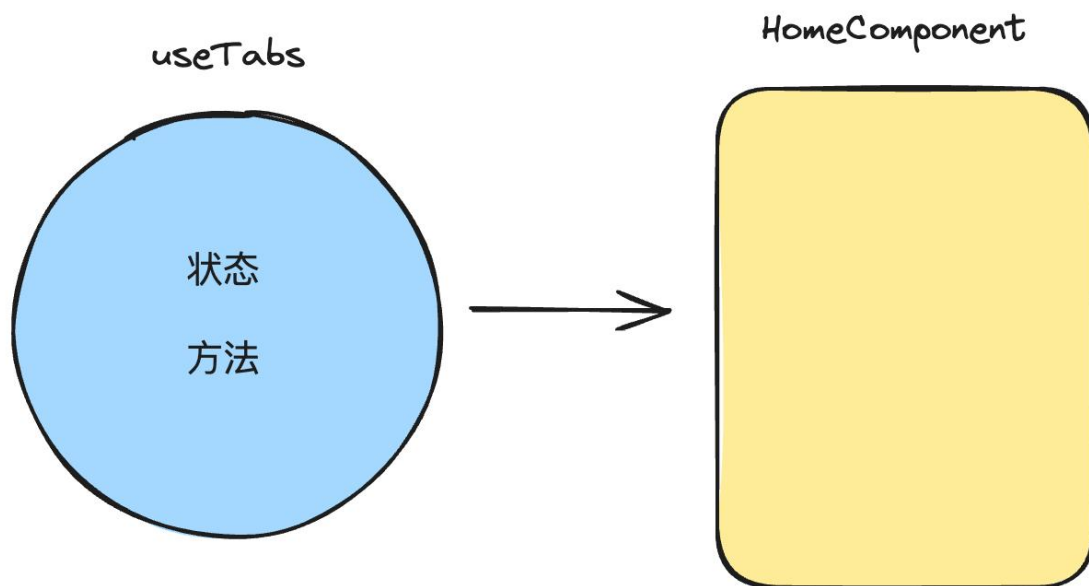
1. 使用 ant-mobile 组件库中的 Tabs 组件进行页面结构创建
2. 使用真实接口数据进行渲染
3. 有优化的点进行优化处理

Home模块-自定义hook函数优化（可选）

场景：当前状态数据的各种操作逻辑和组件渲染是写在一起的，可以采用自定义hook封装的方式让逻辑和渲染相分离

实现步骤：

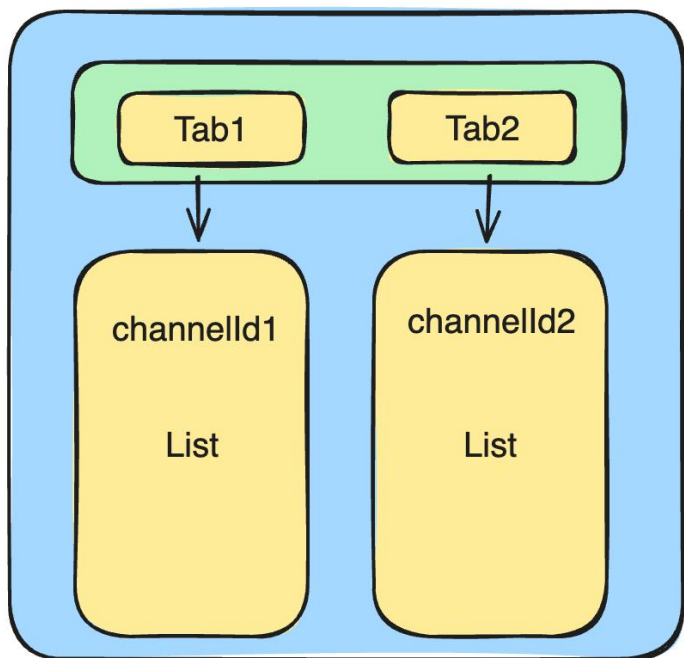
1. 把和Tabs相关的响应式数据状态以及操作数据的方法放到hook函数中
2. 组件中调用hook函数，消费其返回的状态和方法



Home模块-List组件实现

实现步骤：

1. 搭建基础结构，并获取基础数据
2. 为组件设计 channelId 参数，点击tab时传入不同的参数
3. 实现上拉加载功能



Home模块-List列表无限滚动实现

交互要求：List列表在滑动到底部时，自动加载下一页列表数据

实现思路：

1. 滑动到底部触发加载下一页动作

`<InfiniteScroll/>`

2. 加载下一页数据

`pre_timestamp` 接口参数

3. 把老数据和新数据做拼接处理

`[...oldList, ... newList]`

4. 停止监听边界值

`hasMore`

详情模块-路由跳转&数据渲染

需求：点击列表中的某一项跳转到详情路由并显示当前文章

1. 通过路由跳转方法进行跳转，并传递参数
2. 在详情路由下获取参数，并请求数据
3. 渲染数据到页面中

