哔哩哔哩移动端 - 项目实战

前言

今年8月， Vue3.2 终于定稿了 <script setup> 语法糖，作者尤雨溪自信放话： <script setup> + TS + Volar = 真香。

今晚，给大家分享最新的 Vue3.2 + TS 入门到项目实战，做一个B 站移动端项目。   
格局打开，拥抱技术未来！

项目预览

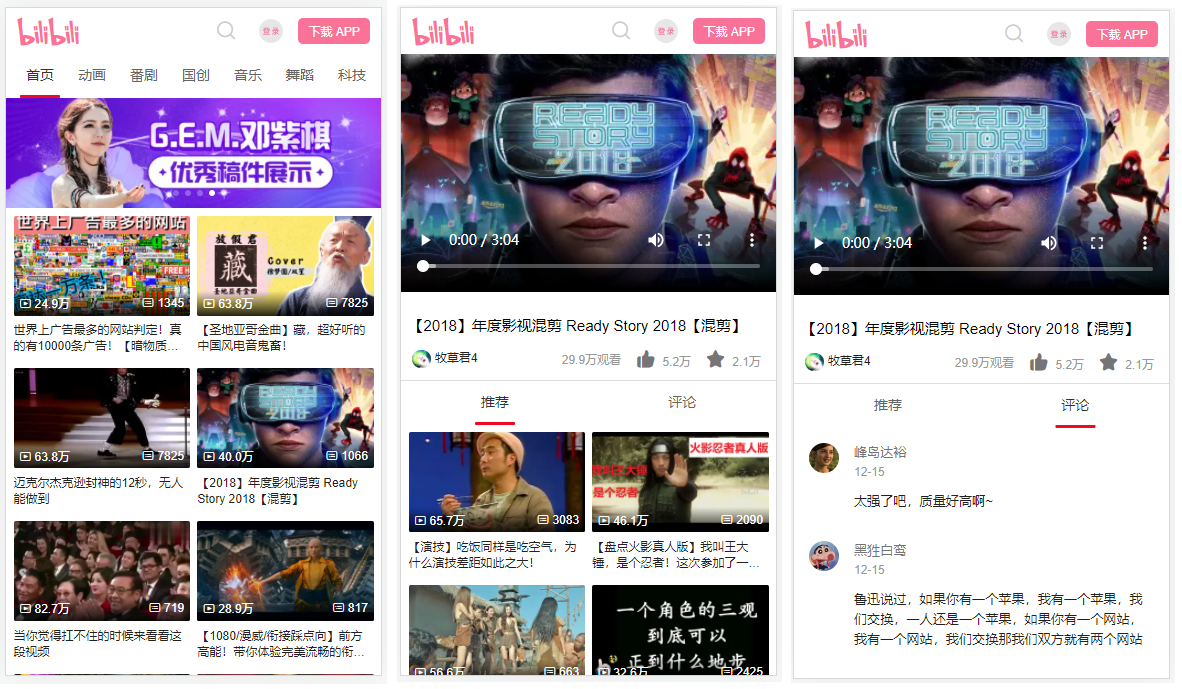
在线预览

<http://javascript.zbztb.cn/tsbilibili/#/>

扫码预览



项目截图



直播收获

1. 完成带数据交互的B站移动端；

2. 学习到 Vue3 + TypeScript 的开发环境配置 Volar + DevTools ；   
3. 学习到 Vant3 组件库 + Axios 请求库在项目实战中使用。

用到的技术

1. Vue Cli <https://cli.vuejs.org/zh/guide/installation.html>

2. Vue3.2 <https://v3.cn.vuejs.org/>

3. VueRouter4 <https://next.router.vuejs.org/zh/guide/>

4. Vant 3 组件库 <https://vant-contrib.gitee.io/vant/v3/#/zh-CN>   
5. TypeScript <https://www.typescriptlang.org/zh/play>

6. Axios 请求库

7. Volar 插件

8. VueDevtools 调试工具

Vue3 + TypeSciprt 开发环境准备 

VScode 安装 Volar

Volar 的最新更新，目前连插件图标都还没制作好 (可以理解，哈哈)



Chrome 安装 vue3\_devtools

1. 浏览器地址栏输入 chrome://extensions/   
2. 右上角开发者模式 - 打开开关。

3. 把 vue3\_devtools.crx 拖放进去安装。

创建 Vue3+TypeScript 项目

使用 @vue/cli 创建项目

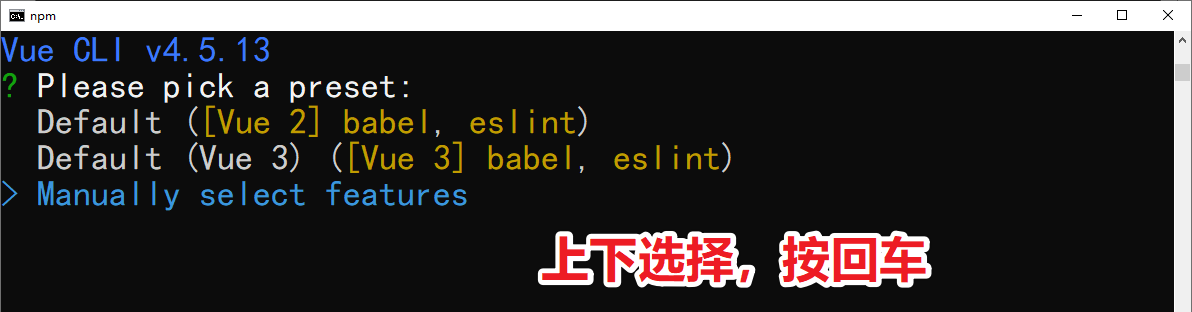
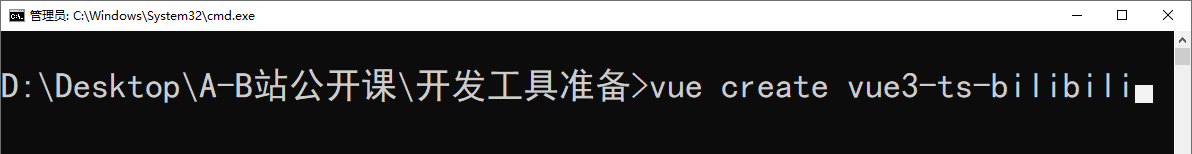
打开终端工具，输入 vue create 项目名称 创建 Vue3 + TS 项目

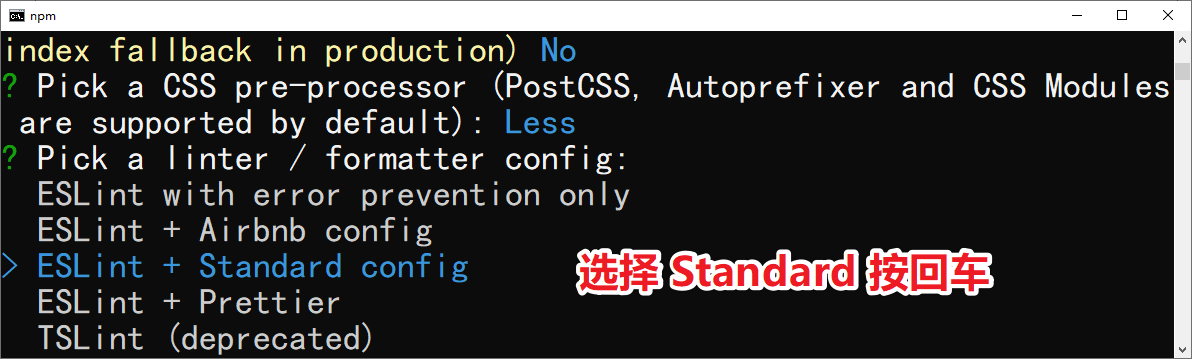
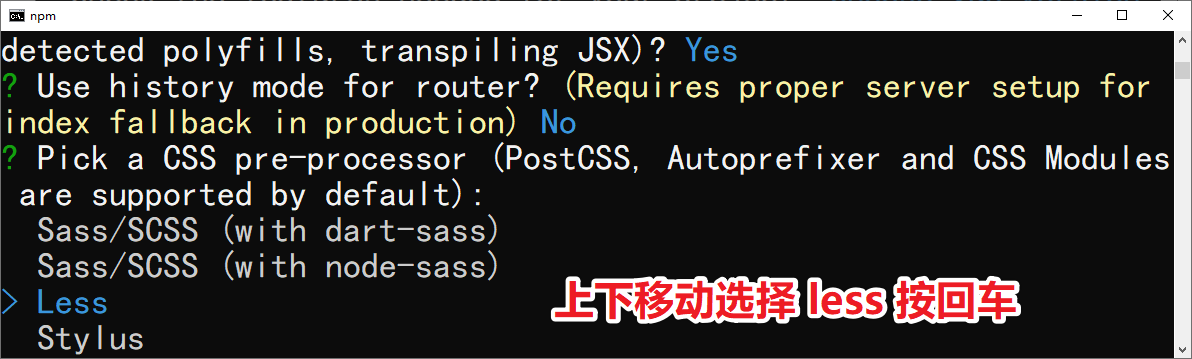
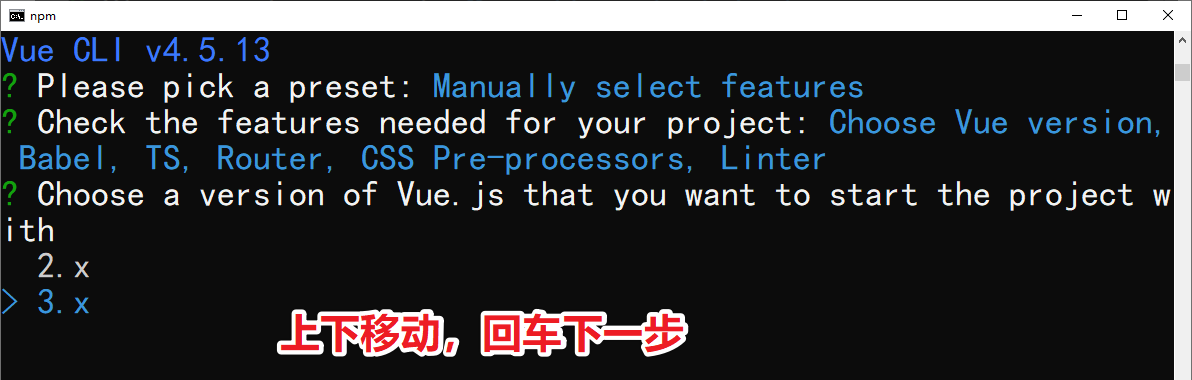
vue create 项目名称

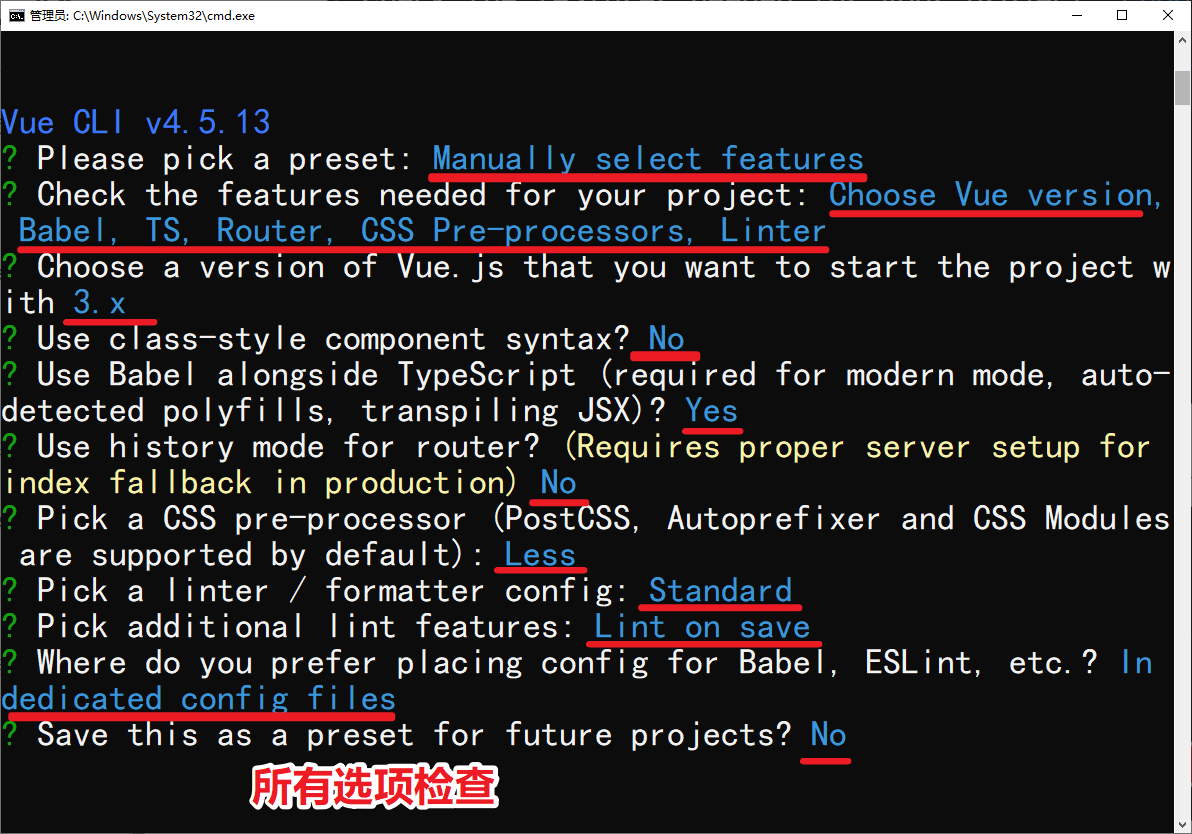
🔔 温馨提醒：

如果创建失败，需要提前安装 @vue/cli 脚手架工具，才能通过以上命令创建项目。

如何安装 @vue/cli 官方教程：<https://cli.vuejs.org/zh/guide/installation.html>

选择 Vue3 + TypeScript 配置 





下载成功提示



清理项目多余文件

操作步骤

删除 assets 文件夹下所有文件

删除 components 文件夹下所有文件   
删除 views 文件夹下所有文件

新建 Home 首页 和 Video 视频播放页   
修改 src\router\index.ts 文件内容   
修改 src\router\index.ts 文件内容

新建两个页面

新建首页 src\views\Home\index.vue

<template>

 <h1>Bilibili 主页</h1>

 <router-link to="/video/1">点我去视频详情页</router-link> </template>

新建视频详情页 src\views\Video\index.vue

<template>

 <h2>视频详情页</h2>

 <router-link to="/">点我回首页</router-link> </template>

修改 src\router\index.ts 文件内容

注意：前面新增了项目的两个页面，我们需要添加路由配置才能看到页面哦。

🔔温馨提醒：+ 表示新增的代码，- 表示删除的代码

import { createRouter, createWebHashHistory, RouteRecordRaw } from 'vue-router' -import Home from '../views/Home.vue'

const routes: Array<RouteRecordRaw> = [

+ { path: '/', component: () => import('@/views/Home/index.vue') },

+ { path: '/video/:id', component: () => import('@/views/Video/index.vue') }

- {

-   path: '/',

-   name: 'Home',

-   component: Home

- },

- {

-   path: '/about',

-   name: 'About',

-   // route level code-splitting

-   // this generates a separate chunk (about.[hash].js) for this route

-   // which is lazy-loaded when the route is visited.

-   component: () => import(/\* webpackChunkName: "about" \*/ '../views/About.vue') - }

]

const router = createRouter({   
 history: createWebHashHistory(), routes

})

export default router 

修改 src\router\index.ts 文件内容

<template>

- <div id="nav">

-   <router-link to="/">Home</router-link> |

-   <router-link to="/about">About</router-link> - </div>

<router-view/>

</template>

-<style lang="less">

-#app {

- font-family: Avenir, Helvetica, Arial, sans-serif; - -webkit-font-smoothing: antialiased;

- -moz-osx-font-smoothing: grayscale;

- text-align: center;

- color: #2c3e50;

-}

-#nav {

- padding: 30px;

- a {

-   font-weight: bold;

-   color: #2c3e50;

-   &.router-link-exact-active {   
-     color: #42b983;

-   }

- }

-}

-</style>

运行 Vue3 + TS 项目

升级到 Vue3 最新版本

打开终端工具，输入 npm i vue@next 即可安装 Vue3 最新版本。

npm i vue@next

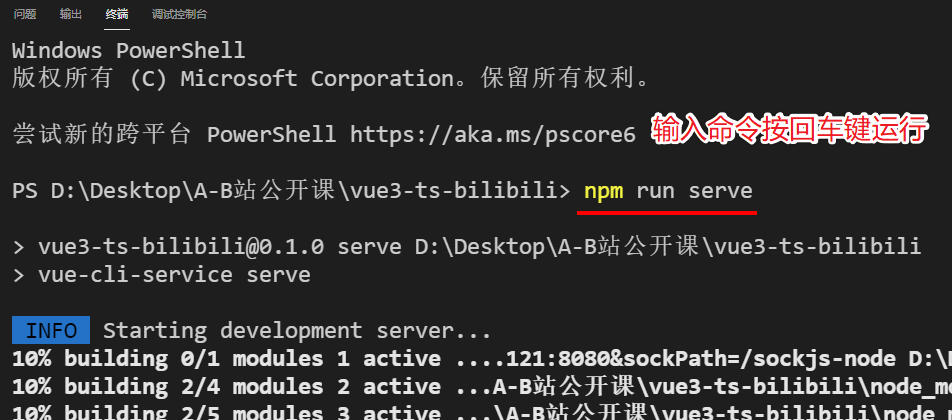


项目运行命令

打开终端工具，输入 npm run serve 即可运行项目。

npm run serve

运行截图



运行成功提示



项目素材准备

前言

哔哩哔哩移动端项目提供了 图片素材 和 样式文件

操作步骤

把素材文件夹中的 assets 文件夹移动到项目中，覆盖项目的 src\assets 文件夹。   
项目入口素材样式

项目入口文件： src\main.ts

import { createApp } from 'vue' import App from './App.vue'   
import router from './router'

+import '@/assets/styles/base.less'   
+import '@/assets/styles/iconfont.less'

const app = createApp(App)

app.use(router)   
app.mount('#app')

项目首页

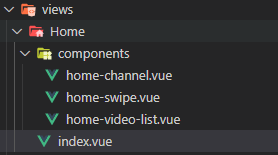
前言

首页按组件划分功能模块。

页面组件分析 

新建首页三个组件备用

频道组件： src\views\Home\components\home-channel.vue   
轮播组件： src\views\Home\components\home-swipe.vue   
视频列表组件： src\views\Home\components\home-video-list.vue   
温馨提醒：新建三个文件备用即可，组件里面暂时不需要写入内容。



首页 - 头部组件

组件素材准备

头部组件为公共组件，主要是静态结构。

哔哩哔哩移动端项目提供了 头部组件 和 单个视频组件 的静态结构素材。

把素材文件夹中的 components 文件夹移动到项目中，覆盖项目的 src\components 组件文件   
夹。

修改项目首页

组件素材复制到项目后，在项目首页中即可使用头部组件。   
文件路径： src\views\Home\index.vue

<template>

- <h1>Bilibili 主页</h1>

- <router-link to="/video/1">点我去视频详情页</router-link> + <!-- 头部组件 -->

+ <AppHeader />

</template>

+<script setup lang="ts">

+// script setup 只需要导入组件，无需注册

+import AppHeader from '@/components/app-header.vue' +</script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何导入组件和使用组件啦，为自己鼓掌。👏 

首页 - 频道组件

前言

前端领域有很多成熟的组件库可以提高我们的开发效率，频道模块我们使用组件库快速实现。⚡   
Vant 是有赞前端团队开源的移动端组件库， Vant 官方提供了 Vue3 版本 ，感谢有赞前端团队

的贡献。🎉

安装组件库

打开终端工具，输入 npm i vant@next 即可安装， vant@next 表示安装最新版 Vant 3 。

npm i vant@next

配置按需引入组件 - 重点难点💥

移动端项目用不上所有的 Vant 组件，全部引入会让项目变大，会导致用户浏览速度收到严重影   
响。

🔔我们的项目是按需引入 Vant 组件，但是需要额外的配置才可以。

打开终端工具，安装 babel-plugin-import 插件。

npm i babel-plugin-import -D

在项目 babel.config.js 中添加配置

module.exports = {

+ plugins: [

+   [

+     'import',

+     {

+       libraryName: 'vant',

+       libraryDirectory: 'es',

+       style: true

+     }

+   ]

+ ],

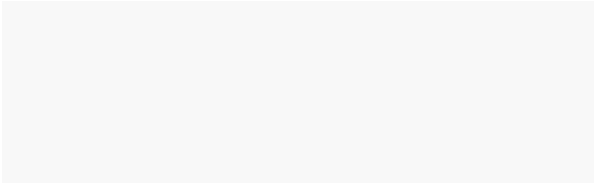
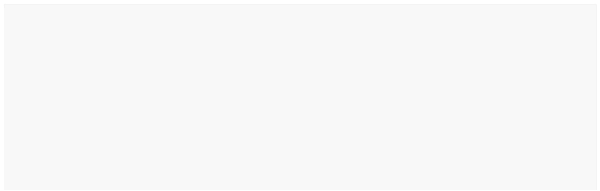
presets: [

  '@vue/cli-plugin-babel/preset' ]

}

频道组件使用 Tab 组件

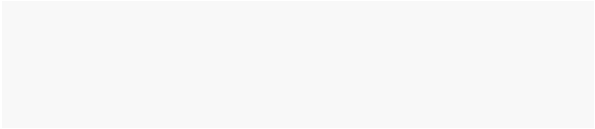
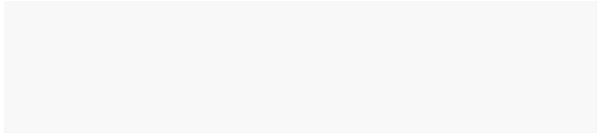
组件库使用前需要在 src\main.ts 中全局按需导入和注册。

import { createApp } from 'vue' import App from './App.vue'   
import router from './router'

import '@/assets/styles/base.less'   
import '@/assets/styles/iconfont.less'

+// 导入Vant组件

+import { Tab, Tabs } from 'vant'

const app = createApp(App) +// 注册Vant组件   
+app.use(Tab)   
+app.use(Tabs)

app.use(router)   
app.mount('#app')

新建频道组件： src\views\Home\components\home-channel.vue ，添加以下代码。

<template>

 <van-tabs v-model:active="active">

   <van-tab title="标签 1">内容 1</van-tab>    <van-tab title="标签 2">内容 2</van-tab>    <van-tab title="标签 3">内容 3</van-tab>    <van-tab title="标签 4">内容 4</van-tab>  </van-tabs>

</template>

<script setup lang="ts">

// ref 函数用于定义模板中使用的响应式数据，相当于 Vue2 的 data import { ref } from 'vue'

// active 表示当前选中标签的下标为 0

const active = ref(0)

</script>

在首页中使用频道组件： src\views\Home\index.vue

<template>

<!-- 头部组件 --> <AppHeader />

+ <!-- 频道组件 --> + <HomeChannel /> </template>

<script setup lang="ts">

// script setup 只需要导入组件，无需注册

import AppHeader from '@/components/app-header.vue'   
+import HomeChannel from './components/home-channel.vue' </script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何使用 Vant3 组件库的组件啦，为自己鼓掌。👏 

首页 - Axios 数据请求

前言

项目通过 mockjs 通过数据服务。   
项目通过 axios 请求数据。

安装 mockjs

打开终端工具，安装 mockjs 插件 和 @types/mockjs TS语法支持包。

npm i mockjs @types/mockjs  -D

把素材文件夹中的 mock 文件夹移动到项目中，用于提供数据服务，切勿遗漏该步骤。💥 在项目入口文件中 src\main.ts 添加导入，提供 mock 数据服务。

import { createApp } from 'vue'

import App from './App.vue'

import router from './router'

import '@/assets/styles/base.less'   
import '@/assets/styles/iconfont.less' import { Tab, Tabs } from 'vant'   
+import '@/mock/index'

const app = createApp(App) app.use(Tab)

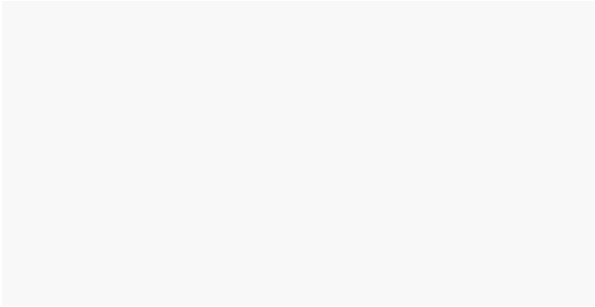
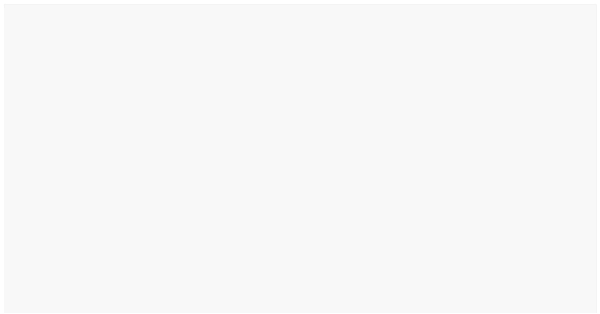
app.use(Tabs)   
app.use(router)   
app.mount('#app')

安装 axios

打开终端工具，安装 axios 用于请求数据。

npm i axios

发送请求获取数据



<template>

<van-tabs v-model:active="active">

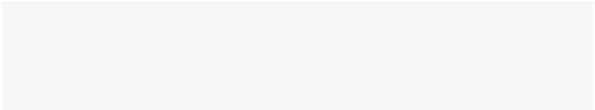
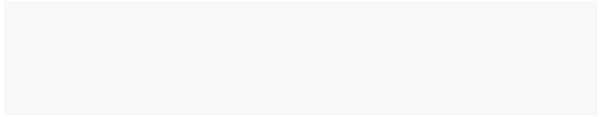
  <van-tab title="标签 1">内容 1</van-tab>   <van-tab title="标签 2">内容 2</van-tab>   <van-tab title="标签 3">内容 3</van-tab>   <van-tab title="标签 4">内容 4</van-tab> </van-tabs>

</template>

<script setup lang="ts">   
import { ref } from 'vue' +import axios from 'axios' const active = ref(0)

+axios({

+ url: '/navList',

+ method: 'get'

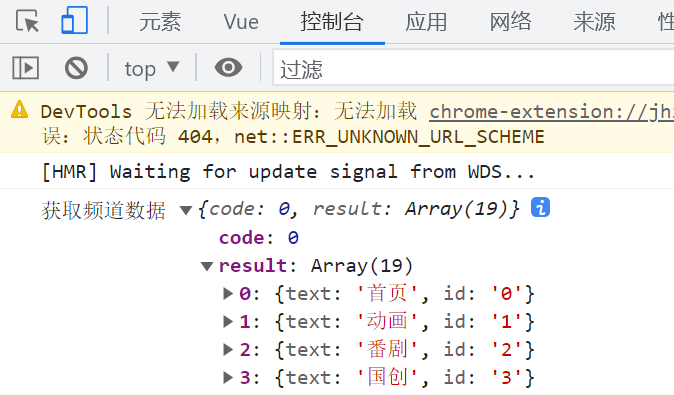
+}).then(res => {

+ console.log('获取频道数据', res.data) +})

</script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何使用 Axios 发送请求获取数据啦，再为自己鼓掌。👏



首页 - 频道组件数据渲染

前言

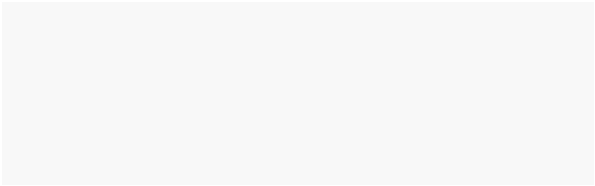
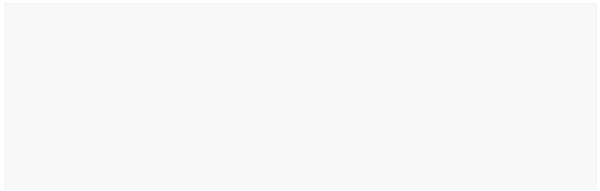
TypeScript 的 interface 接口定义用于标记数据格式

接口定义好处：模板中使用 变量 的时候，鼠标悬停变量有类型提示，书写代码也有提示

通过 TypeScript 设置数据的类型，再大型的项目会更好维护。🎉

渲染频道组件数据

组件路径： src\views\Home\components\home-channel.vue



<template>

<van-tabs v-model:active="active">

+   <van-tab v-for="item in list" :key="item.id" :title="item.text"></van-tab> -   <van-tab title="标签 1">内容 1</van-tab>

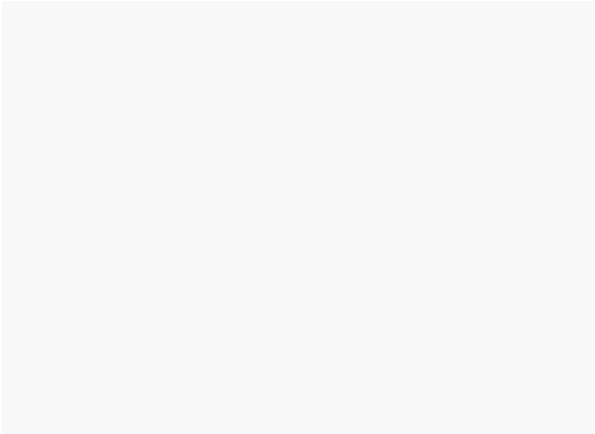
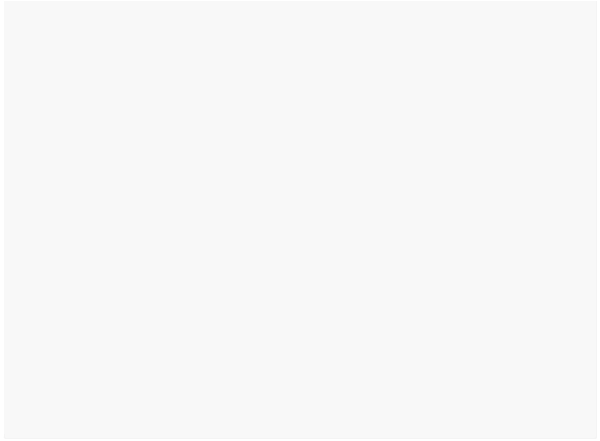
-   <van-tab title="标签 2">内容 2</van-tab>

-   <van-tab title="标签 3">内容 3</van-tab>

-   <van-tab title="标签 4">内容 4</van-tab>

</van-tabs>

</template>

<script setup lang="ts"> import { ref } from 'vue' import axios from 'axios'

+// TypeScript 的接口用于标记数据格式 +interface INavItem {

+ id: string

+ text: string

+}

const active = ref(0)

+// 频道数据， <INavItem[]> 表示 list 数据为数组，数组的每一项需要复合 INavItem 接口的格式 +// TypeScript 好处：模板中使用 list 和 item 的时候，鼠标移入有类型提醒

+const list = ref<INavItem[]>([])

axios({

url: '/navList',

method: 'get'

}).then(res => {   
 console.log('获取频道数据', res.data) + list.value = res.data.result

})

</script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何使用 TypeScript 的接口定义数据格式了，再为自己鼓掌。 👏



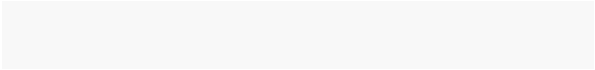
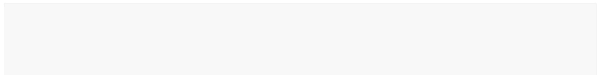
首页 - 轮播图组件

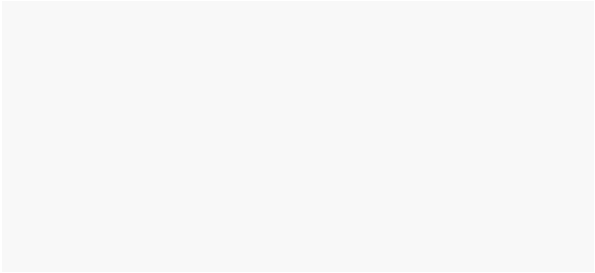
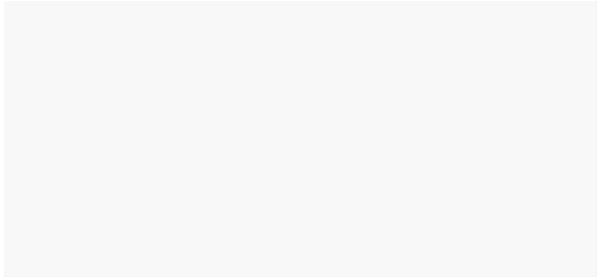
前言

首页轮播图需要使用到 Vant3 组件库的 Swipe 和 SwipeItem 组件。   
轮播图请求接口为 { url: '/swiperList', method: 'get' } 。   
可参考频道组件实现，建议各位小伙伴自己实操完成哦。

落地代码

项目入口导入和注册组件 src\main.ts

import { createApp } from 'vue' import App from './App.vue'   
import router from './router'

import '@/assets/styles/base.less'   
import '@/assets/styles/iconfont.less' import '@/mock/index'

-import { Tab, Tabs } from 'vant'

+import { Tab, Tabs, Swipe, SwipeItem } from 'vant'

const app = createApp(App) app.use(Tab)

app.use(Tabs)   
+app.use(Swipe)   
+app.use(SwipeItem)   
app.use(router)   
app.mount('#app')

轮播组件代码参考： src\views\Home\components\home-swipe.vue

<template>

 <van-swipe class="my-swipe" :autoplay="3000" indicator-color="white">    <van-swipe-item v-for="item in list" :key="item.imgSrc">

     <img :src="item.imgSrc" />

   </van-swipe-item>

 </van-swipe>

</template>

<script setup lang="ts"> import { ref } from 'vue' import axios from 'axios'

interface ISwiper {   
  imgSrc: string   
  link: string

}

const list = ref<ISwiper[]>([])

axios({

 url: '/swiperList',

 method: 'get'

}).then(res => {

 console.log('轮播图数据', res.data.result)  list.value = res.data.result

})

</script>

<style lang="less" scoped> .my-swipe {

 img {

   width: 100%;

}

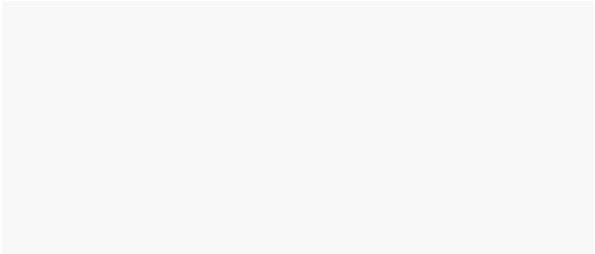
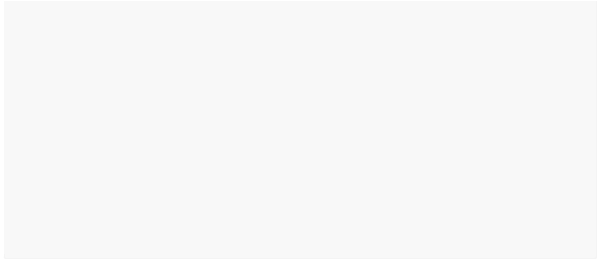
}

</style>

轮播组件在首页导入： src\views\Home\index.vue



<template>

 <!-- 头部组件 --> <AppHeader />   
 <!-- 频道组件 --> <HomeChannel />

+ <!-- 轮播组件 --> + <HomeSwipe />   
</template>

<script setup lang="ts">

import AppHeader from '@/components/app-header.vue'   
import HomeChannel from './components/home-channel.vue' +import HomeSwipe from './components/home-swipe.vue'   
</script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何使用 Vant3 + Axios + TypeScirpt 组合使用啦 ，你的格   
局已经打开，为自己努力学习鼓掌三次。👏👏👏   
喜欢我整理的文档吗？我是黑马程序员广州前端的苏老师哦，欢迎线下和我见面交流，比心。🤞   
PS：我也喜欢喝奶茶。



B站防盗链问题

前言

可能有些小伙伴的轮播图看不到图片，主要是 B 站的图片做了防盗链处理（我们能理解）。   
我们可以添加代码避免出现这种情况（我们能通过技术破解访问限制）😎

操作步骤

打开项目文件： public\index.html ，添加以下代码即可修复问题。

<!DOCTYPE html>

<html lang="">

<head>

  <meta charset="utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width,initial-scale=1.0"> +   <meta name="referrer" content="no-referrer">

  <link rel="icon" href="<%= BASE\_URL %>favicon.ico">

  <title><%= htmlWebpackPlugin.options.title %></title>

</head>

首页 - 视频列表组件

前言

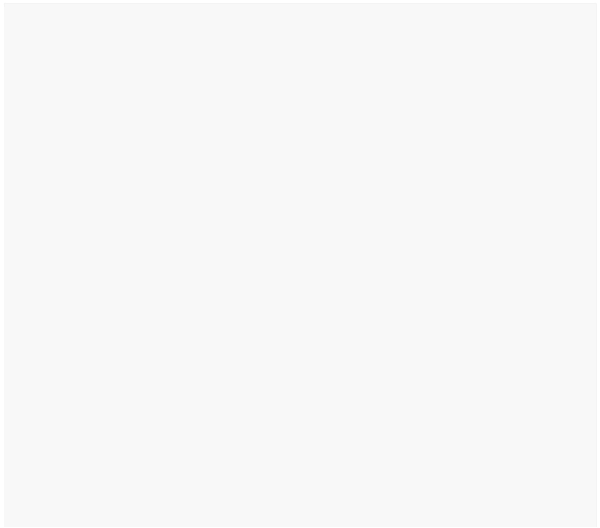
视频列表组件设计到数据父传子

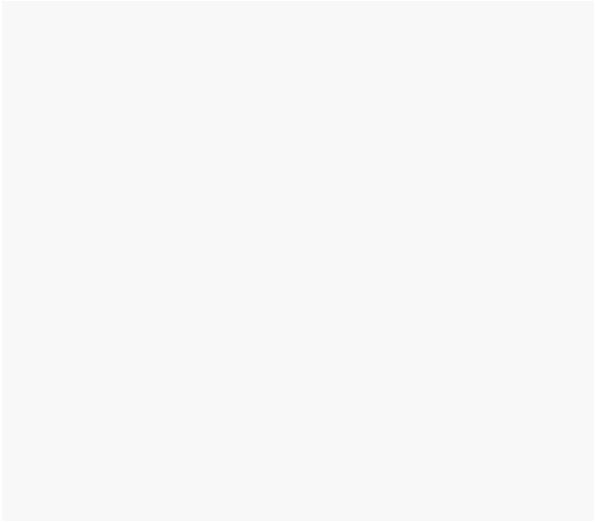
Vue3 中子组件可通过 defineProps 定义 props ，用于接受父组件传递过来的数据。

Vue3 中子组件可通过 PropType 规定 props 数据的接口类型 。

落地代码

视频列表组件代码参考： src\views\Home\components\home-video-list.vue   
<template> 模板中的 :video="item" 表示把数据传递给子组件。



<template>

 <div class="list">

   <AppVideoItem v-for="item in list" :key="item.id" :video="item" />  </div>

</template>

<script setup lang="ts">

import { ref } from 'vue'

import axios from 'axios'

import AppVideoItem from '@/components/app-video-item.vue'

interface IVideoItem {  id: number;

 imgSrc: string;   
  desc: string;   
  playCount: string;   
  commentCount: string;  videoSrc: string;

}

const list = ref<IVideoItem[]>([])

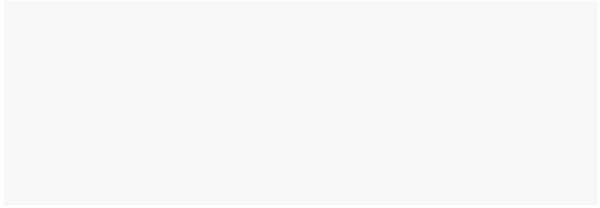
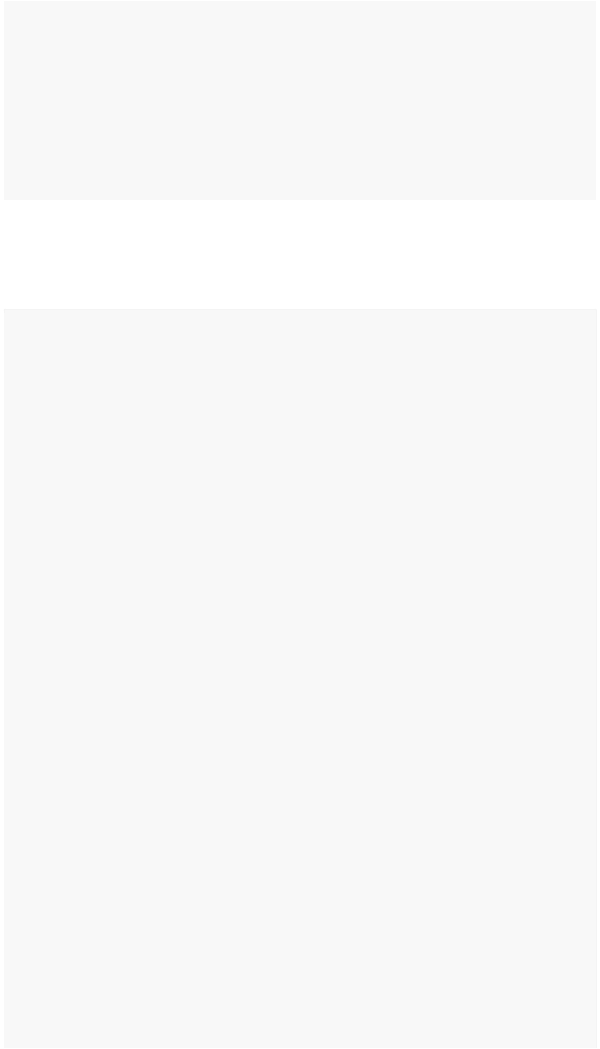
axios({

 method: 'get',

 url: '/videosList'

}).then(res => {

 console.log('视频列表的数据', res.data.result)  list.value = res.data.result

})

</script>

<style lang="less"> .list {

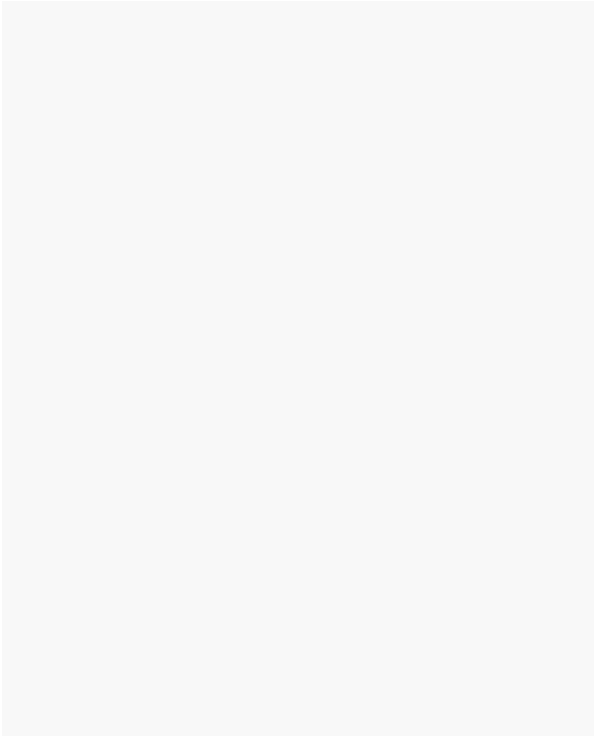
 display: flex;   
  flex-wrap: wrap;  padding: 0 1vw;

}

</style>

视频子组件 src\components\app-video-item.vue 接受父组件传过来的数据   
Vue3 中通过 defineProps 接收数据，可通过 PropType 定义数据的接口类型。

温馨提醒：通过 TypeScript 定义了接口，在 <template> 模板中调用数据的时候会有代码提示 哦。🧐

<template>

- <router-link class="v-card" :to="`/video/0`">

+ <router-link class="v-card" :to="`/video/${video.id}`">

  <div class="card">

    <div class="card-img">

-       <img class="pic" src="@/assets/images/loading.png" alt="视频描述" /> +       <img class="pic" :src="video.imgSrc" :alt="video.desc" />

    </div>

    <div class="count">

      <span>

        <i class="iconfont icon\_shipin\_bofangshu"></i>

+         {{ video.playCount }}

      </span>

      <span>

        <i class="iconfont icon\_shipin\_danmushu"></i>

+         {{ video.commentCount }}

      </span>

    </div>

  </div>

-   <p class="title">当你觉得扛不住的时候来看看这段视频</p>

+   <p class="title">{{ video.desc }}</p>

</router-link>

</template>

<script setup lang="ts">

+import { defineProps, PropType } from 'vue'

+interface IVideoItem { + id: number;

+ imgSrc: string;

+ desc: string;

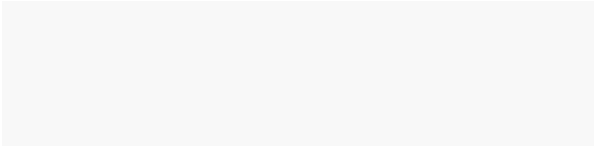
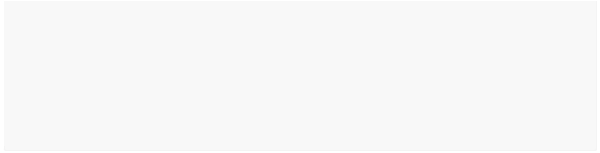
+ playCount: string;

+ commentCount: string; + videoSrc: string;

+}

+// 🔔 父传子需要通过 defineProps 接收 +// 🔔 PropType 用于指定数据格式   
+defineProps({

+ video: {

+   // Object 对象为 IVideoItem 接口格式

+   type: Object as PropType<IVideoItem>, +   required: true

+ }

+})

</script>

视频列表组件在首页导入： src\views\Home\index.vue

<template>

<!-- 头部组件 -->   
 <AppHeader />   
 <!-- 频道组件 -->   
 <HomeChannel />   
 <!-- 轮播组件 -->   
 <HomeSwipe />

+ <!-- 视频列表 -->

+ <HomeVideoList /> </template>

<script setup lang="ts">

import AppHeader from '@/components/app-header.vue'

import HomeChannel from './components/home-channel.vue'

import HomeSwipe from './components/home-swipe.vue'

+import HomeVideoList from './components/home-video-list.vue' </script>

🌈目标验证

恭喜你已经学会在 Vue3 项目中如何使用 defineProps + PropType 处理父子组件通讯业务了。 到此为止，你通过努力把整个哔哩哔哩首页完成，是不是感觉自己越来越棒棒哒，全体黑马前端人 员给你点赞。👍 x 10086



视频播放页 - 自我实战

前言

有了前面哔哩哔哩首页的实战经验，视频播放页建议大家自己书写哦。💪   
我们页提供了视频播放页的素材文件，练习过程中遇到困难的小伙伴，可以去参考哦。🧐   
自我实战的时候慢一点没关系哈，关键是坚持到最后，偷偷地努力，最后惊艳身边所有人。🐮🍺

页面组件分析



最后

收获

通过你的不懈努力，你已经成功收获：

1. 完成带数据交互的B站移动端；

2. 学习到 Vue3 + TypeScript 的开发环境配置 Volar + DevTools ；

3. 学习到 Vant3 组件库 + Axios 请求库在项目实战中使用。

感言

为自己的努力点个赞吧，也为黑马前端用心提供的直播分享点个赞，一键三连。👍   
如果你觉得很棒，可以分享给你身边的小伙伴，大家一起学习会更快乐哦。😘

黑马前端 B 站账号，前端程序员聚集地：<https://space.bilibili.com/415434293>   
本次直播来自广州黑马前端苏老师，欢迎各位小伙伴来广州黑马吃喝玩乐。（PS：我能教你们几句 粤语哦）