

Vue学习第6天

反馈

回顾

实在无法创建项目的解决方案（重要）

1. vue create 做的事情
 1. 创建项目的文件结构
 2. 帮我们装了npm包，第三方模块
 3. git相关的初始化
 - ...
2. 可以copy同桌已经生成好的脚手架项目
 1. 把node_modules删除再copy
 2. 在copy过来的项目目录下执行npm install

```
1 | cd 项目目录
```

3. 运行 cd项目目录 npm run serve

Vue-cli项目结构

1568512436611

Vue-cli src代码结构

1. main.js是页面的入口
 1. el选择器和public/index.html是关联的
 2. Vue.config.productionTip = true 显示生成模式信息
 3. 创建了Vue根实例
 4. 在根实例内部渲染了主组件App.vue
2. App.vue主组件，页面内容都是由App.vue渲染的
3. assets 静态资源文件夹
4. components 组件文件夹，除了 App.vue 之外的组件，都写到这个文件夹中即可

Vue-cli项目编码位置

1. 组件的逻辑直接写在 xx.vue
2. 静态资源放到 assets 文件夹下面，直接使用对应路径即可引入
3. css也放在 assets 这个文件夹，如何引入
 1. style标签中引入

```
1  /* 使用css支持的语法导入 */
2  /* @import url('./assets/base.css'); */
```

2. main.js 中引入

```
1  // 导入 样式
2  import './assets/base.css'
```

组件的全局与局部注册

1. 全局注册。一次注册，到处使用

```
1  vue.component(组件名,{组件属性})
```

2. 局部注册。哪个组件注册，就只能在那个组件里面使用

```
1 new Vue({
2   components:{
3     组件名:{组件属性}
4   }
5 })
```

注意点：

1. 一个html里面可以有多个Vue实例

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-
7   scale=1.0" />
8   <meta http-equiv="X-UA-Compatible" content="ie=edge" />
9   <title>Document</title>
10
11 </head>
12
13 <body>
14   <div id="app">
15     <h2>实例1</h2>
16     <news></news>
17     <counter></counter>
18   </div>
19   <div id="app2">
20     <h2>实例2</h2>
21     <news></news>
22     <counter></counter>
23   </div>
24   <script src="../lib/vue.js"></script>
25   <script>
26     vue.component('news', {
27       template: `<div>全局注册</div>`
28     })
29     const app = new Vue({
30       el: "#app",
31       data: {},
```

```
30     components: {
31       counter: {
32         template: `

## 你点击了3次</h2>` 33 } 34 } 35 }); 36 const app2 = new Vue({ 37 el: "#app2", 38 data: {} 39 }); 40 </script> 41 </body> 42 43 </html>


```

Vue-cli项目的组件注册

2. 全局注册组件，在main.js引入注册

3. 引入组件 import 组件对象 from '地址'

2. Vue.component(名字，组件对象)

3. 在任意组件里面，可以把组件名当标签进行使用

4. 局部注册组件，在需要用到这个组件的地方引入

1. import 组件对象 from '地址'

2. 把组件对象设置给components属性

3. 在组件里面可以把组件名当标签进行使用

5. 全局的组件就全局注册，loading,dialog; 哪个组件需要就引入; 大部分时候都是局部注册

组件的name属性

Vue官方推荐每个组件都给一个name属性

给一个name的话，那么Vue开发工具可以看到组件的名字。

Vue-cli项目的路由整合(练习)

准备工作

1. 创建或者复制一份脚手架项目
2. 复制的项目需要，cd 项目目录，`npm install`，安装第三方模块
3. `npm run serve` 运行项目

整合路由

1. 装包：在工程目录下执行，也就cmd, 执行一个命令。`npm install vue-router`
2. 导包：`import VueRouter from 'vue-router'`
3. 用包
 1. `Vue.use(VueRouter)` 明确地安装路由功能
 2. 创建路由规则
 1. 创建一个组件 `xxx.vue`
 2. 引入组件
 3. `routes=[{path:"/xx",component:组件}]`
 3. 创建路由对象
 1. `router`
 4. 设置给Vue实例
 1. `new Vue({ router })`

编码位置

1. 导入 注册路由 `main.js`
2. `routerlink router-view` `app.vue`
3. 添加组件 `components/`
4. 静态资源 `assets`

注意

1. `npm run serve`时会占用命令行，一般停掉，再安装好包好后，再执行`npm run serve`启动项目
2. 安装包的名字一般去包的官网去找，一般都有

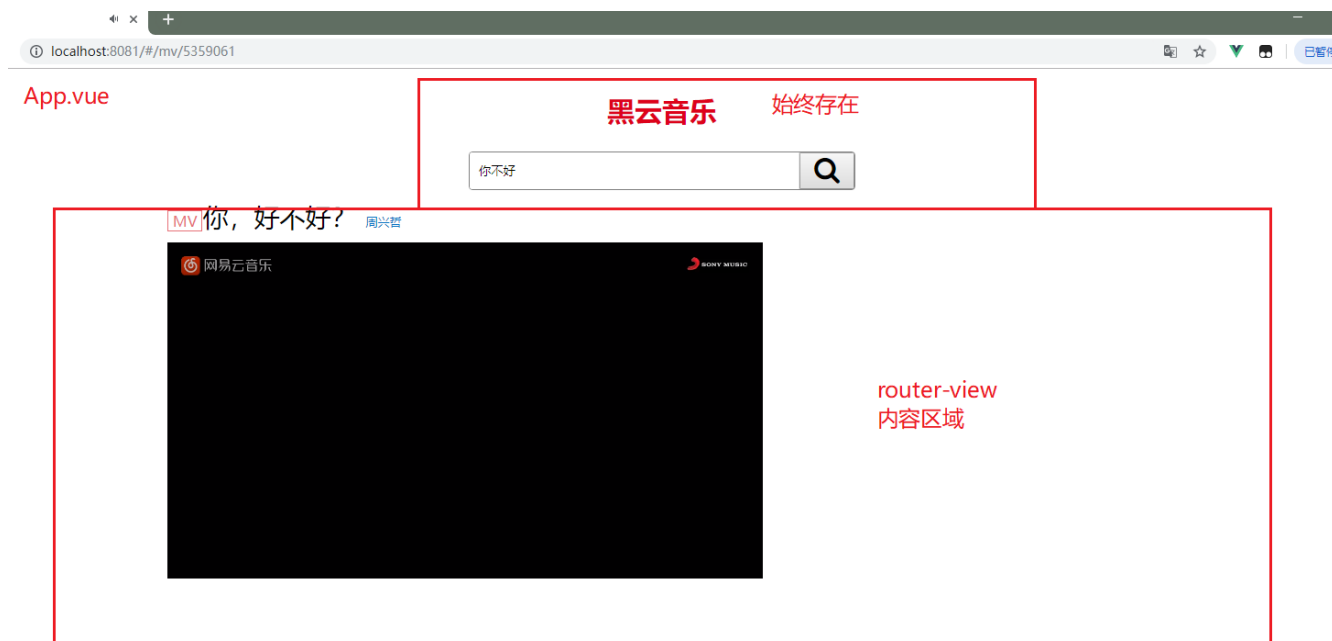
1. 目前这个项目里面只是引入了vue.js对吧，我们昨天还学习一个重要的Vue插件叫作路由，那么我们把路由也整合进vuecli项目。

2. 我们新建一个项目，不在前一个项目的里面改动。copy一个项目运行成功。
3. 看vue-router官网，如何安装
4. 由于需要把路由实例传递给Vue实例，那么导入注册路由应该放在main.js
5. 菜单和内容是页面内容部分，应该放在App.vue里面

player-界面分析

1. 项目准备

1. 新建一个02.player-cli文件夹
2. copy usevue-router的项目，除了node_modules
3. 在02.player-cli下npm install
4. npm run serve运行 02.player-cli项目



player-搜索区域整合

1. copy index.html到App.vue
2. 在App.vue导入样式
3. 添加router-view

player-搜索区域抽取为组件

1. 新建components/search.vue
2. 局部注册组件

player-轮播图

轮播图组件及路由

1. 创建轮播图组件 components/Slider.vue
2. main.js中导入组件
3. main.js添加路由规则
 1. path: "/slider"
 2. component: slider

注意

1. 添加了路由规则之后, 修改url上的hash, 验证轮播图组件及路由是正确的。

路由重定向

[传送门](#)

当访问hash1时, 跳转到hash2 (想象成电话的呼叫转移)

使用方法

```
1  const routes = [{
2    path: '/',
3    redirect: '/slider'
4  }, {
5    path: '/slider',
6    component: Slider
7  }]
```

注意点

重定向的地址 如果没有对应组件, 页面会显示空白

饿了么ui 介绍

[传送门](#)

1. 饿了么前端团队开发的pc端的基于vue的组件库
2. 内部封装了很多现成的组件，直接就可以使用比如轮播图，tab栏，省市联动，开关，table
3. 因为在vue开发时用的很多，所以和几个其他的库合并到一起被称之为 **vue全家桶**
 1. vue
 2. axios
 3. vue-router
 4. 饿了么ui
 5. vuex

轮播图组件使用

1. 安装包 `npm i element-ui -S`
2. 引入element-ui

```
1 import ElementUI from 'element-ui';  
2 import 'element-ui/lib/theme-chalk/index.css';  
3 Vue.use(ElementUI);
```

3. copy走马灯示例代码到Slider.vue
4. 尽早发请求初始化轮播图组件
 1. created请求接口初始化
 2. axios.get调用接口<https://autumnfish.cn/banner>
 3. 要用axios
 1. 安装axios
 2. 引入
 3. 使用
 4. 渲染页面上 拿到数据 banners 结合vfor渲染
 5. 可能还需要调整轮播图的样式

注意点

1. 饿了么ui的轮播图模板默认提供的是h3标签，需要自行替换为别的标签
2. 默认有一个高度 150px，可能根据需求调整
3. 组件一出现就要调用接口，一般放到哪个钩子
 1. created:这里可以访问data

总结
