

《Vue基础》第十三单元

路由 / vue-router



路由是什么？

- URL (Uniform Resource Locator) ,
统一资源定位
- SPA (single-page application) ,
单页应用

路由的本质，是访问路径和
页面视图的一一对应关系

单页应用 / Vue 的路由

- hash 模式，使用 URL hash 值
- history 模式，HTML5 History API

— # 之隔

两种模式对比

hash 模式	history 模式
<code>/#/home</code> <code>/#/singer</code> <code>/#/singer/003CoxJh1zFPpx</code>	<code>/home</code> <code>/singer</code> <code>/singer/003CoxJh1zFPpx</code>
简单实用 支持所有浏览器	优雅清爽 依赖 HTML5 History API 依赖服务器配置

注：URL 路径均省略了域名前缀 <http://m.jxsjs.com>

hash 模式路由的原理

```
mounted() {  
  window.onhashchange = () => {  
    let { hash } = window.location;  
    hash = hash.substring(hash.indexOf('#') + 2);  
    for (let key in this.visible) {  
      this.visible[key] = false;  
    }  
    this.visible[hash] = true;  
  };  
}
```

课堂案例：hash 模式路由实现原理演示

vue-router

- Vue 官方路由管理器，和 Vue 深度集成
- 支持 hash 模式和 history 模式
- 基于组件，可嵌套路由，路由传参
- 链接导航，编程式导航

使用 vue-router

1. 使用 router-view 渲染路径匹配到的视图组件
2. 导入 vue 和 vue-router 及各个组件
3. Vue.use(Router)
4. 定义路由配置，每个路由映射一个组件
5. 创建 router 实例，传入路由配置
6. 将 router 实例注入 vue 根实例

使用 vue-router

导入 vue 和
vue-router
及各个组件

```
import Vue from 'vue';  
import Router from 'vue-router';  
import Home from '@views/Home.vue';  
import Singer from '@views/Singer.vue';
```

安装 vue-router

```
Vue.use(Router);
```

定义路由配置
每个路由映射
一个组件

```
const routes = [{  
  path: '/home',  
  component: Home  
}, {  
  path: '/singer',  
  component: Singer,  
}];
```

创建 router 实
例，传入路由配
置

```
const router = new Router({  
  routes  
});  
  
export default router;
```


使用 vue-router

router 实例
注入
vue 根实例

```
import Vue from 'vue';  
import router from './router';
```

```
new Vue({  
  router,  
  render: h => h(App)  
}).$mount('#app');
```

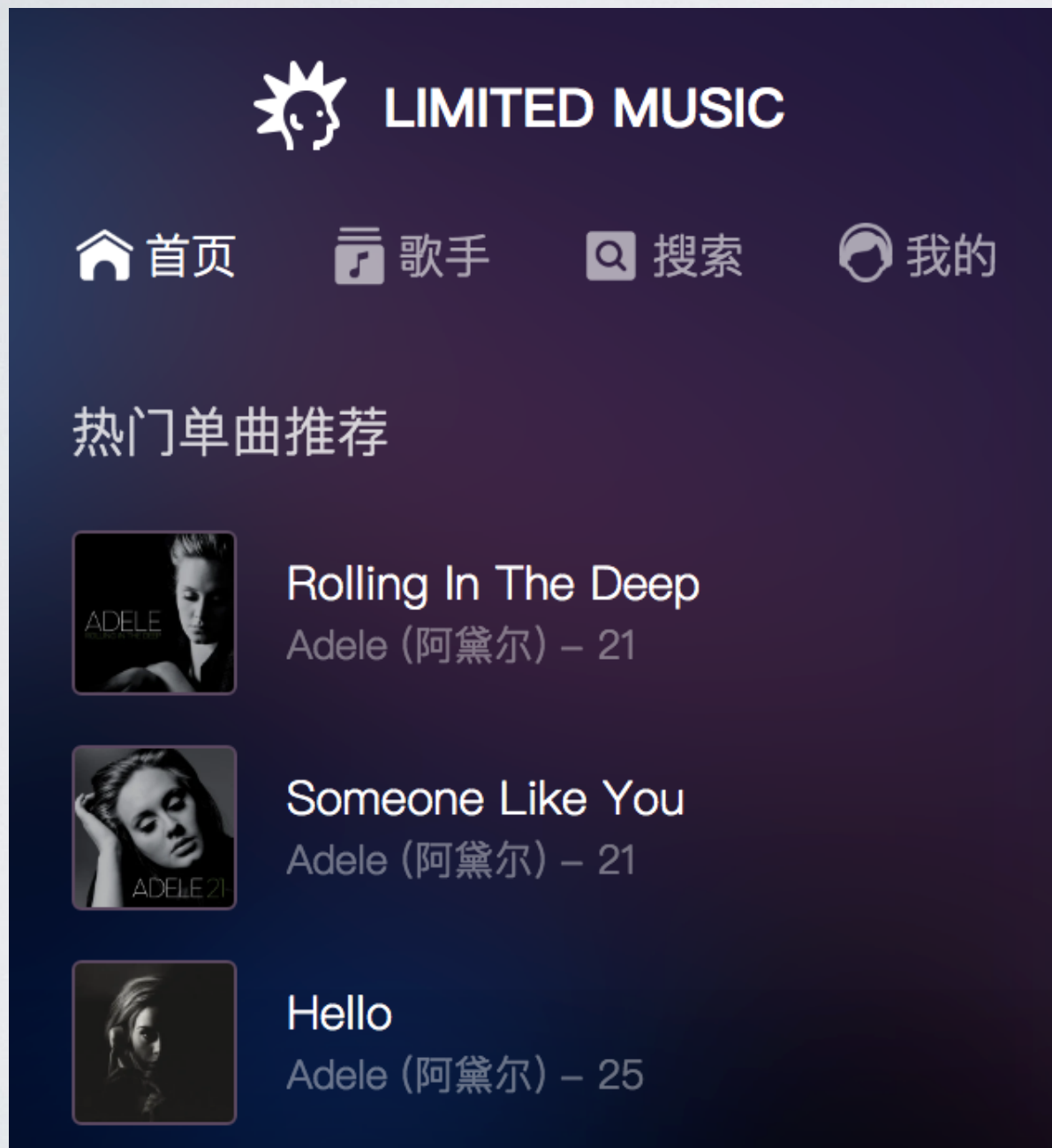
\$mount 可以理解为
指定 el 挂载点

render 函数可以暂时理解为
定义 template 的一种方法

案例

课堂案例：播放器路由

- 首页 /home
- 歌手列表 /singer
- 搜索 /search
- 我的 /user



导航

- 使用 **router-link** 创建 a 标签
- 借助 router 实例方法，编程导航

```
router-link(  
  v-for="(d, i) in navList"  
  :key="i"  
  tag="div" 使用其他标签  
  :to="d.link"  
) {{ d.label }}
```

```
this.$router.push(`/singer/${id}`);
```

课堂案例：实现 my-router-link 组件

导航

- router.replace / router-link(replace)
- router.go(n)

```
· this.$router.go(1); // 等同于 history.forward()  
· this.$router.go(-1); // 等同于 history.back()
```

传参和取参

- 使用 **path** 拼接
- 使用 **params** 和 **query** 传参
- 使用当前路由 **this.\$route** 取参

注意与 \$router 的区别

```
· router-link(:to="***)  
· router.push(***)
```

传参

```
· router-link(:to="***)  
· router.push(***)
```

如果提供了 path
params 会被忽略
query 则不会

```
· this.$router.push('/singer');  
· this.$router.push({ path: '/singer' });  
· this.$router.push({ name: 'singer', params: { singerId: '007' } });  
· this.$router.push({ path: '/singer', query: { order: 'age' } });
```

```
{  
  path: '/singer/:singerId',  
  name: 'singer',  
  component: SingerDetail  
}
```

动态路径参数
命名路由

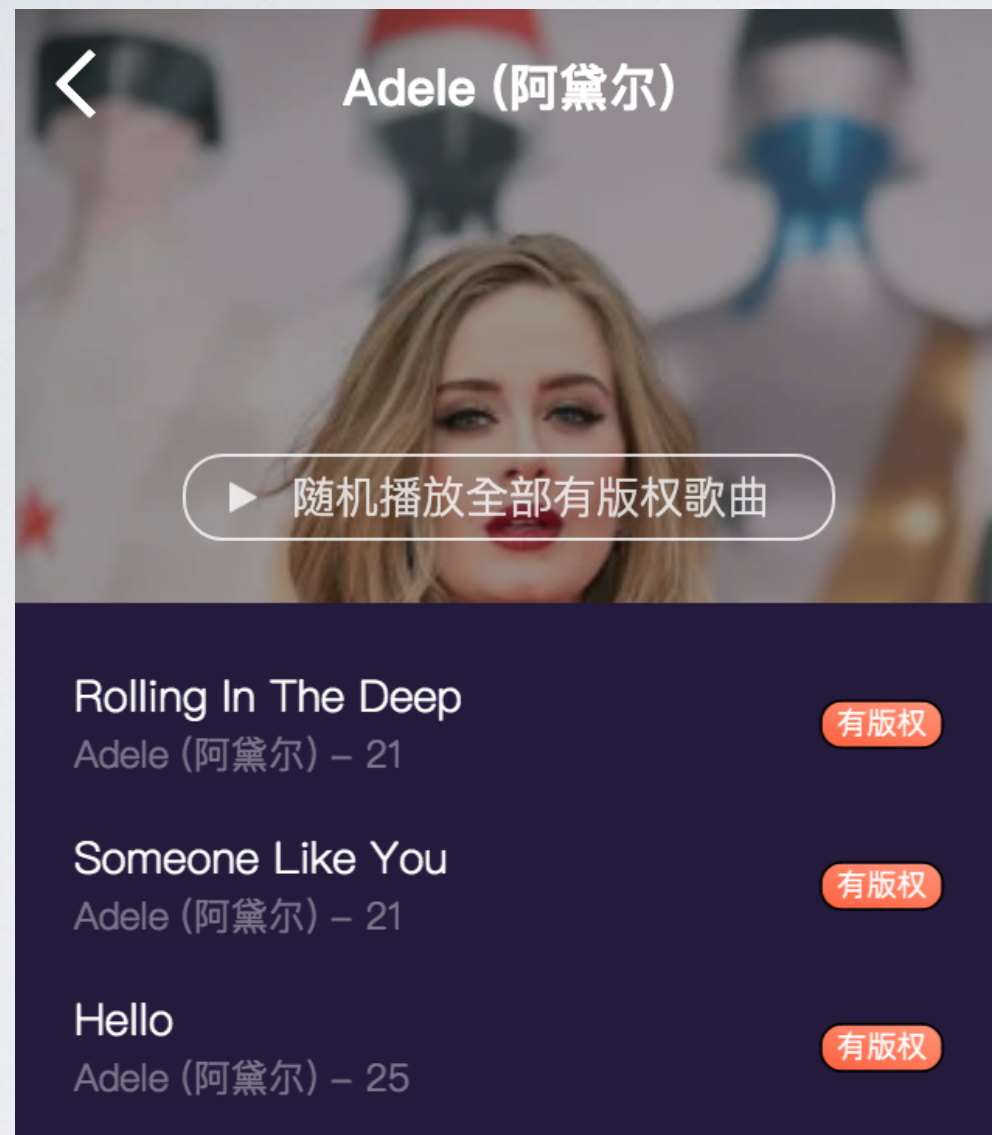
取参

- `this.$route.params`
- `this.$route.query`

```
const { params, query } = this.$route;  
const { singerId } = params;  
const { order } = query;
```

注意这里是 `$route` (当前匹配的路由)
不是 `$router` (整个路由实例)

案例



课堂案例： 歌手详情页参数传递

- 歌手列表 `/singer`
- 歌手详情 `/singer/:singerId`

总结

- 使用 router-view 渲染路径匹配到的视图组件
- 定义路由和组件的映射、创建路由实例、注入 vue 根实例
- 两种导航：router-link 和 router 实例方法
- 传参和取参 \$route.params、\$route.query

课后作业 1

- 删除 limited-music 文件夹，重新创建项目，不要默认安装 vue-router
- 创建好项目后，用 npm 命令手动安装 vue-router
- 定义路由，实现课堂案例

创建项目时，
这个步骤不要
勾选Router

```
? Check the features needed for your project:
  ☒ Babel
  ☐ TypeScript
  ☐ Progressive Web App (PWA) Support
  ☒ Router
  ☒ Vuex
  ☒ CSS Pre-processors
  ☒ Linter / Formatter
  ☐ Unit Testing
```

课后作业 2

- 了解 history 模式的实现原理
- 了解 Vue.use 方法
- 了解 \$mount 方法和 render 函数