Vue2Project DAY05

Axios的封装设计的实现流程

1. 解决如果在项目中有多个地方都需要发送相同类型的请求时，需要重复写相应的url接口地址与参数的问题。

   在src/http文件夹中新建：index.js，导出httpApi对象，在该对象中提供发送请求的接口方法。这样，谁想发请求，只需要引入该httpApi对象，调用方法即可。

   ```
   import myaxios from "./MyAxios"

   const httpApi = {

     /** 查询首页演员列表 */
     queryActors(){
       let url = "http://localhost:3010/movie-actors"
       let params = {page:1, pagesize:100}
       return myaxios.get(url, params)
     }

   }

   export default httpApi
   ```

   当需要访问演员列表时，可以引入该js文件，调用httpApi的接口方法获取结果即可：

   ```
   import httpApi from '@/http/index'

   /** 列出演员列表 */
   listActors() {
       httpApi.queryActors().then(res=>{
           this.actors = res.data.data
       })
   }
   ```

2. 当解决了上述问题后，又发现httpApi对象中需要定义所有的请求接口方法，导致httpApi对象臃肿不堪，方法繁杂，所有接口的访问方法都得在这一个对象里。需要进一步封装：按照业务模块将Api接口方法拆开，放在不同的js文件里。

   ```
   src/http/index.js

   src/http/apis/ActorApi.js
   src/http/apis/DirectorApi.js
   src/http/apis/MovieApi.js
   ...
   ```

   当需要访问这些API方法时，就可以：

```
import httpApi from '@/http/index'
httpApi.actorApi.queryActors(params).then(res=>{
    查询演员
})
httpApi.directorApi.save(params).then(res=>{
    新增导演
})
```

3. 针对业务API子模块，封装请求路径的前缀，使得方便的切换生产环境与测试环境。因为所有的业务接口子模块使用统一的前缀，应该把路径单独的封装到js中，谁用谁引入：

```
src/http/BaseUrl.js
```

```
/** src/http/BaseUrl.js
  * 暴露出请求资源路径的前缀即可 */
const URLENV = {
  DEV: {  // 开发环境
    BMDURL : 'http://localhost:3010',
    UPLOADURL:'http://localhost:9000'
  },
  PRO: {  // 生产环境
    BMDURL : 'https://web.codeboy.com/bmdapi',
    UPLOADURL:'https://web.codeboy.com/bmduploadapi'
  }
}
export default URLENV.DEV
```

电影管理模块

整理左侧边栏导航及路由子页面

设计嵌套路由：

```
/home/movie-list    看到电影列表页
/home/movie-add     看到新增电影页
```

1. 准备两个页面：

```
src/views/movie/MovieList.vue
src/views/movie/MovieAdd.vue
```

2. 配置嵌套路由：

```
/home/movie-list    看到电影列表页
/home/movie-add     看到新增电影页
```

3. 添加左侧边栏菜单，点击后跳转到相应地址。

搭建电影列表静态页面

面包屑导航
搜索表单
表格
分页器

组件挂载完毕时加载电影列表

1. 在组件的mounted中，发送请求，加载电影列表数据。

   MyAxios httpApi MovieApi.js queryMovies()

2. 将服务端返回的数据以el-table的方式显示在页面中。

   movies:[] `<el-table :data="">` `<el-table-column>自定义列模板</el-table-column>`

3. 实现分页业务。

实现模糊查询电影列表

在输入框中填写关键字，敲回车、点查询按钮都可以实现模糊查询。更新列表。

实现电影删除业务

业务需求：点击列表项后的删除按钮，发送请求，将电影信息删除，重新更新列表：

1. 删除完毕后，还要看到当前页的数据。但是被删的电影已经没了。
2. 如果搜索框中有关键，则删除成功后带着关键字一起查询当前页。
3. 如果删除的是最后一页的最后一条，还得加载上一页。

实现思路：

1. 为按钮绑定点击事件，获取选中的电影的ID，作为参数执行事件处理方法。
2. 发请求，执行删除业务。
3. 根据返回的结果处理后续业务。

实现新增电影业务

业务需求：点击新增电影菜单，看到一个表单，填写后点击提交，新增成功。

实现步骤：

1. 准备好一个静态页面，包含一个大表单， 9个表单项。
2. 针对每一个表单项处理收集的数据。
3. 点击提交按钮，实现新增业务。
4. 添加表单验证。