一．启动与启动前的准备

1.安装依赖 npm install

2.开启 npm run serve

二．基础知识

1.前端三层html内容 ,js，css样式逻辑,对应的的是<templatl>,<script>，<style>，后端只需要关注内容与样式部分即可。

2.死的，不变的内容写在内容部分，活的，根据数据变动的内容会放在<script>标签的内的data属性中。了解这个就能够对页面显示的某些固定不变的内容进行修改。

3.调用接口的方法会封装到到api文件夹下的.js文件下。

三．简单的联调操作

1.启动后台服务

2.打开项目

若确认在本地开启后端服务

在vue.Config.js中第8行为

remoteHost = '127.0.0.1'

若被注释掉，需要打开，

第15行为

    target: 'http://127.0.0.1:8082',

3.找到对应要调试的页面。

4.查看这个组件中在templat标签中，标签的v-model属性是和data中的哪个值进行绑定的

。v-model是vue中给出的标签属性，可以实现双向绑定。也就是说，在页面中某个标签下的某个值，其对应绑定的data中的值会改变。同样改变data中的值，标签中与之绑定的值也会发生改变

有了这个概念，就可以看懂ElementUi的文档了。

如：<https://element.eleme.cn/#/zh-CN/component/radio>

5.如果想要封装新的请求方法，就要找到api文件夹下的js文件。这个文件夹分成多个是为了便于管理，比如说查询管理方面的请求会写在searchManage.service.js文件下。

6.封装好了后需要调用。在需要调用的组件内进行引用，引用的方式。如：

import searchService from '@api/searchManage.service.js'

写在export default{}上面

数据请求补充示例（以运营平台为例）

1. 在./src/api 下找到或新建所需要的文件 如’ audit.service.js’
2. 在文件的对应class 中添加方法 ，如：

 methodsName(data) {

    return Service({

      url: '/xxxxx/abcdfg.json',

      method: 'get', //若为post则改为post

      params: data //若为post请求params改为data

    })

  }

3, 在需要此接口的页面引入相应的对象 ,如 步骤中的对象用 export default 输出，可直接

Import ‘定义变量名’ from ‘目标文件地址’ （参见js相关语法）

1. 使用封装好的接口，如步骤3引入的的对象名为‘objName’装的方法名为‘methodsName’

需要传递的参数为data则

可用objName. methodsName(data).then( res=>{ } )

获取返回参数res,共输入到需要的回调函数之中

或者 let res = await objName. methodsName(data)获取返回的数据res ，参考js 的 async/ await 相关语法

5, element-ui 表格组件实例说明

如tableData为以数组 格式为[ {keyA:1,keyB:2}, ………]

<el-table :data="tableData" size="small" border style="width:100%">

 <el-table-column //对应表格中的某列

          prop="keyA" // 对象中的属性名

          label="表头该列的名字"

        />

</el-table>

详见element-ui相关文档