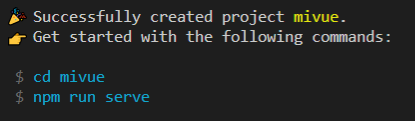
|  |
| --- |
| 下载脚手架  cnpm i -g @vue/cli  下载成功    脚手架下载成功后，创建自己的项目  （1）cd mivue 进入下载的 create vue 项目名  （2）npm run serve 启动脚手架 |



单页面应用

view 下放入页面组件

|  |
| --- |
| (1)页面组件放在view下  (2)配置router.js |

全局组件放在component下 MyHeader.vue

|  |
| --- |
| MyHeader.vue 要变成全局组件 要在main.js中 引入 并注册  (1)引入 import  (2)将MyHeader.vue注册全局组件  Vue.component('my-header', MyHeader)  (3)在App.vue下引入全局组件<my-header><my-header>  写在<router-view>上面 |

下载axios

在项目本地安装 npm i -save axios

main.js下引入axios 共用

|  |
| --- |
| 引入axios  import axios from 'axios'  //注册axios  Vue.prototype.axios=axios  //保存所有axios请求公共的域名部分，今后只要修改，只需要修改一处就好  axios.defaults.baseURL=’http://localhost:3000’  import axios from 'axios'  // 注册axios  Vue.prototype.axios=axios  // 保存所有axios请求公共的域名部分  axios.defaults.baseURL='http://localhost:3000' |
| 组件中使用例如Index.vue 向服务器端发请求  公司讲究封装函数 在assets下引入fun.js    Index.vue   |  | | --- | | 从assets中 引入封装的fun.js函数    method 和created()生命周期钩子函数和methos同级 |   data(){  return{  p1:{},  p2:{}.  p3:{}  other:[]  }  }  methods:{  created(){  this.axios.get(  //必须用箭头函数，使内外this保持一致  'http://localhost:5050/index')  .then(result=>{  console.log(result.data);  //或者对象解构  [this.p1,this.p2,this.p3]=result.data;  this.p1=result.data[0];  this.p2=result.data[1];  this.p3=result.data[2];  this.others=result.data.slice(3);  }  )  } |

# 数据获取

方法一：json-sever 假数据

|  |  |
| --- | --- |
| （1）cmd 安装 全局安装  npm install -g json-server      （2）新建一个文件夹  dbs  db\_index.json 里面放数据  数据格式--json  {‘接口’：  [  {‘属性’：‘属性值’},  {}  ]}    把有主外键关系的列（id）放到最前面  ‘lid’:1     1. 启动json-server 修改端口号 和数据托管的文件名   $ json-server --watch --port 53000 db.json  以下显示成功  测试端口是否有数据    [json-server 详解](https://www.cnblogs.com/fly_dragon/p/9150732.html)  JSON-Server 是一个 Node 模块，运行 Express 服务器，你可以指定一个 json 文件作为 api 的数据源。  **安装json-server**  npm install -g json-server  **启动 json-server**  json-server可以直接把一个json文件托管成一个具备全RESTful风格的API,并支持跨域、jsonp、路由订制、数据快照保存等功能的 web 服务器。  db.json文件的内容：  {  "course": [  {  "id": 1000,  "course\_name": "马连白米且",  "autor": "袁明",  "college": "金并即总变史",  "category\_Id": 2  },  {  "id": 1001,  "course\_name": "公拉农题队始果动",  "autor": "高丽",  "college": "先了队叫及便",  "category\_Id": 2  }  }  }  例如以下命令，把db.json文件托管成一个 web 服务。  $ json-server --watch --port 53000 db.json  输出类似以下内容，说明启动成功。  \{^\_^}/ hi!  Loading db.json  Done  Resources  http://localhost:53000/course  Home  http://localhost:53000  Type s + enter at any time to create a snapshot of the database  Watching...  此时，你可以打开你的浏览器，然后输入：[http://localhost:53000/course](http://localhost:53000/course" \t "_blank) | 全局安装 |

方法二：数据库

|  |
| --- |
|  |

# 获取数据

|  |
| --- |
| 7 |