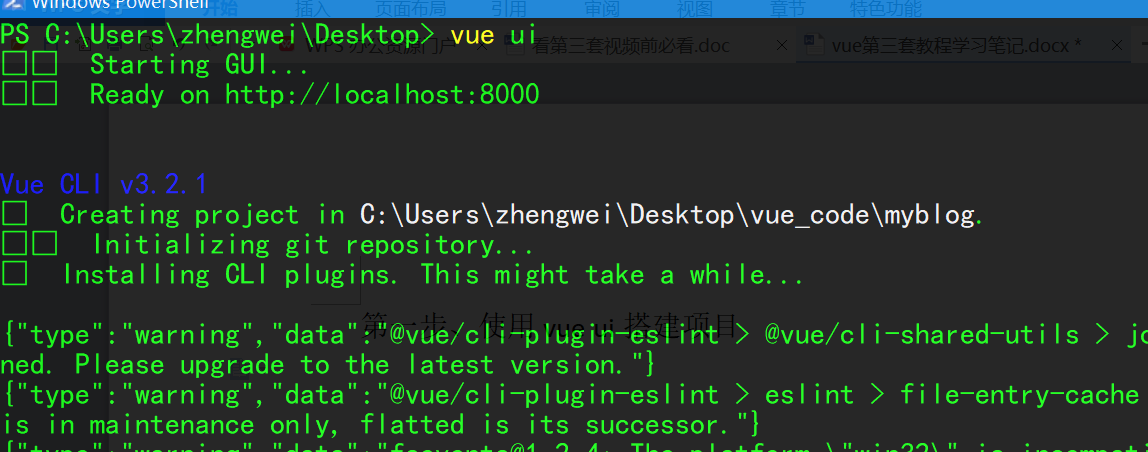
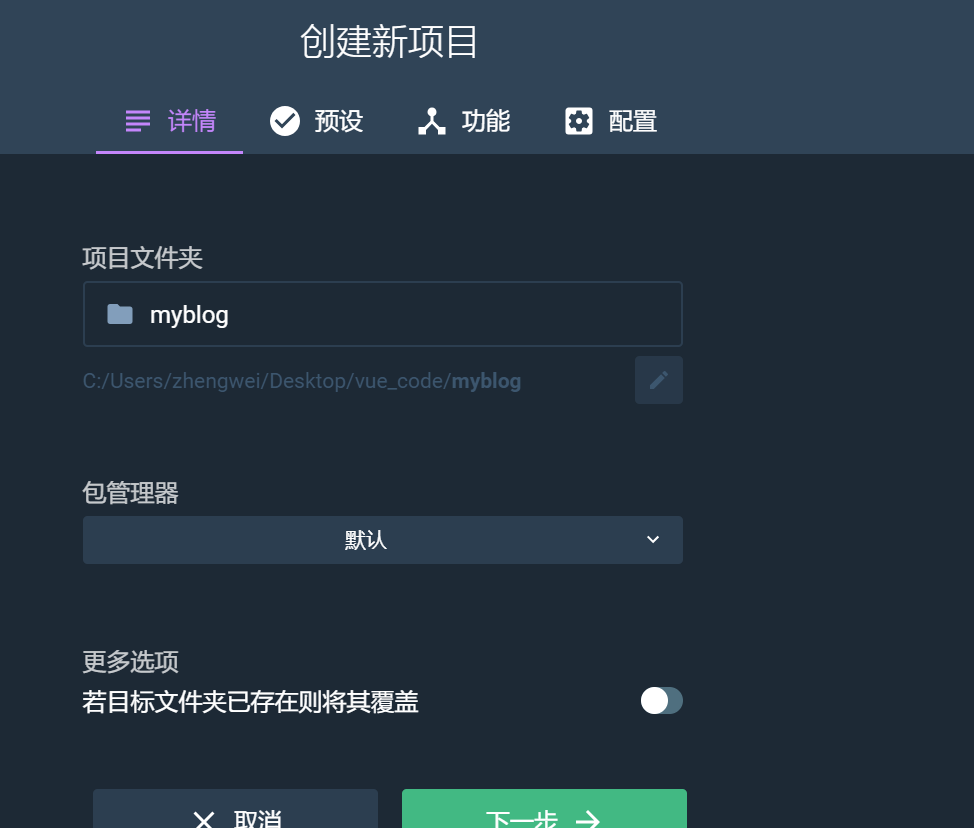
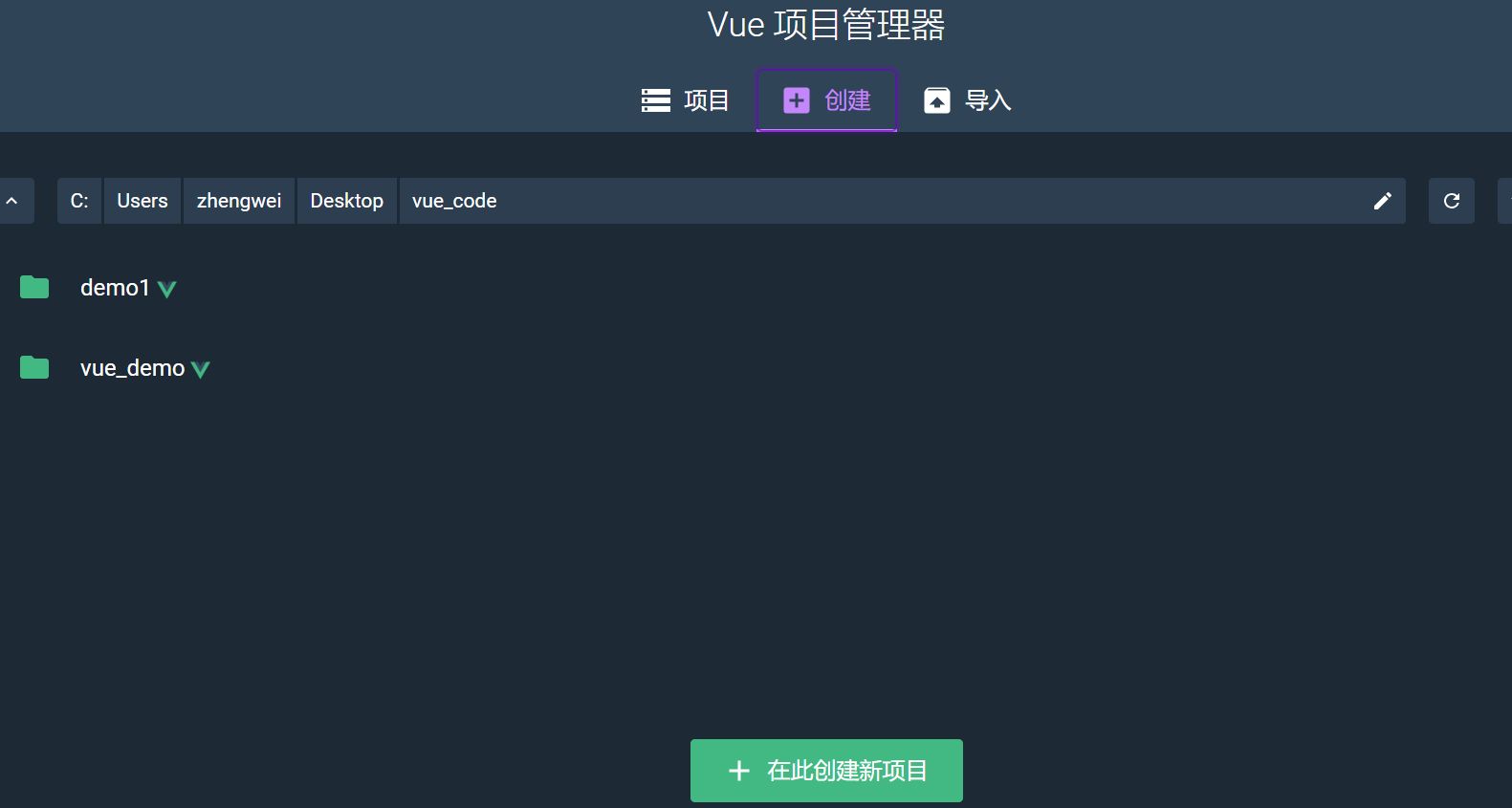
## 注意事项

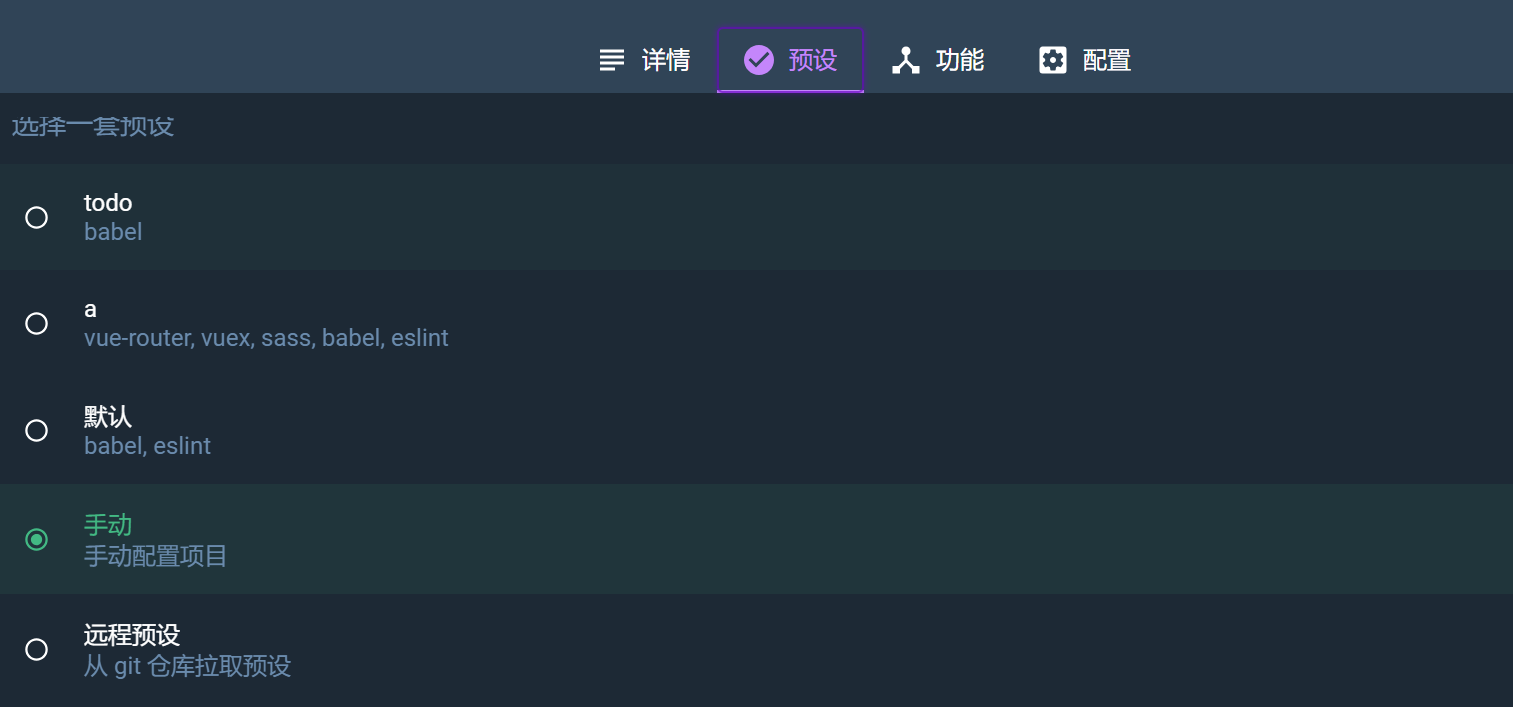
1. 不要用vue-resource,我们自己写的时候替换成axios
2. 样式的细节不要纠结
3. 不要用命令行
4. 不要自己手动设置路由
5. 视频中的目录规划推荐修改为vue ui自动创建的

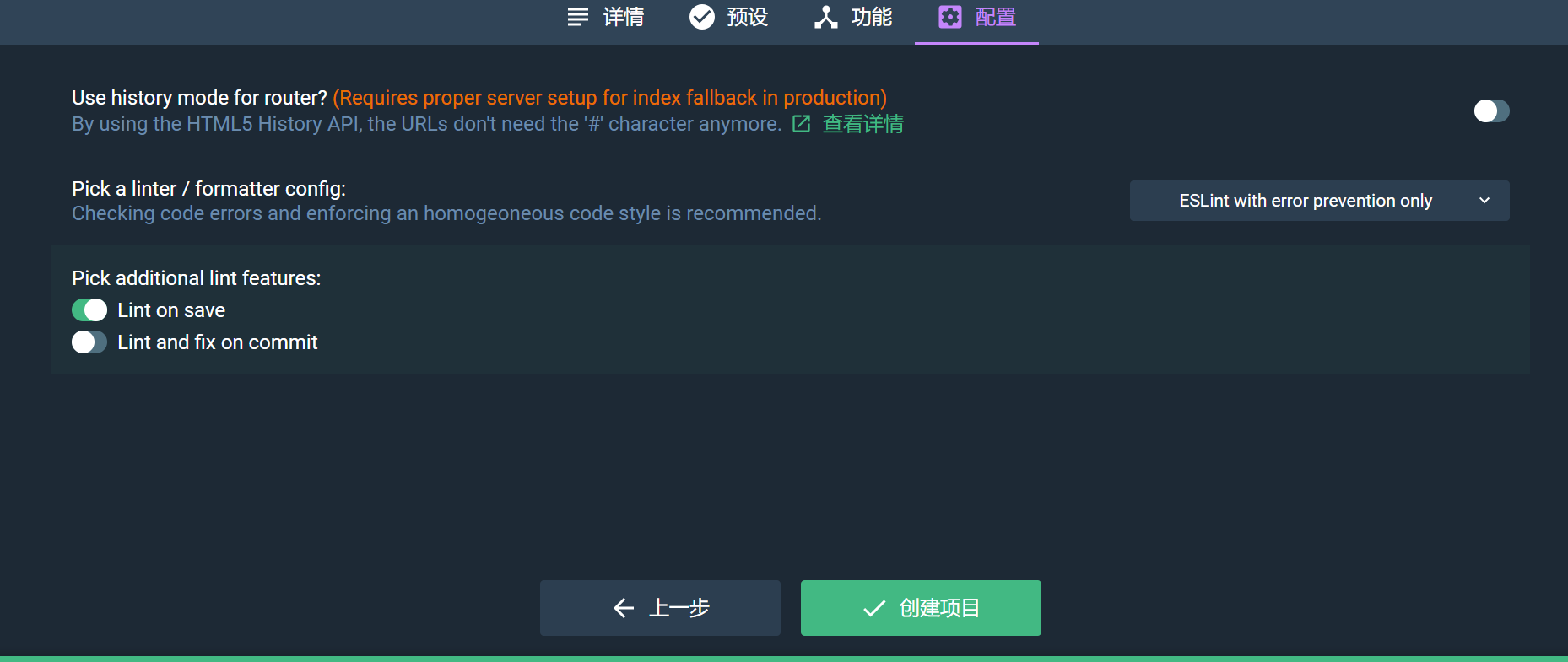
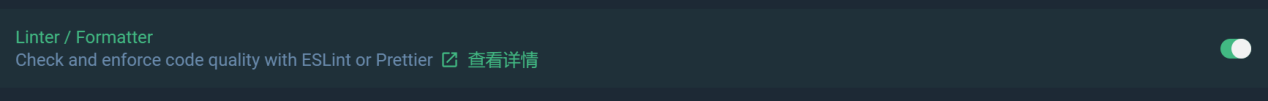
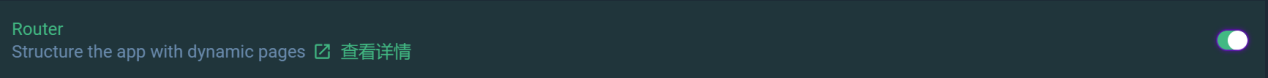
## 搭建项目

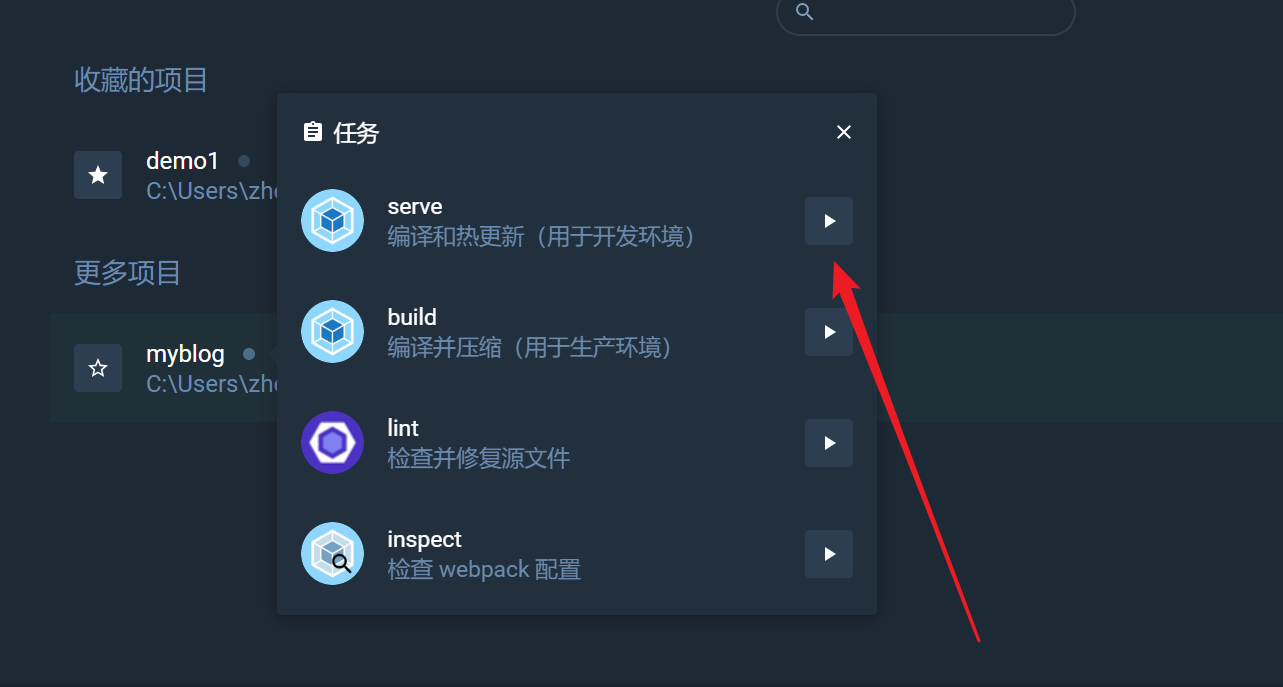
第一步、使用vue ui搭建项目





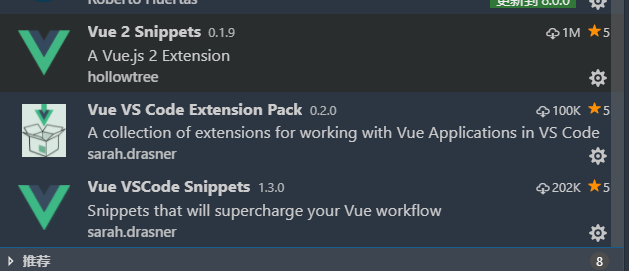


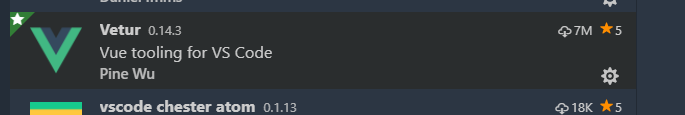




第二步、用vsCode打开项目代码，创建addBlog.vue

vsCode中我安装的vue插件如下：

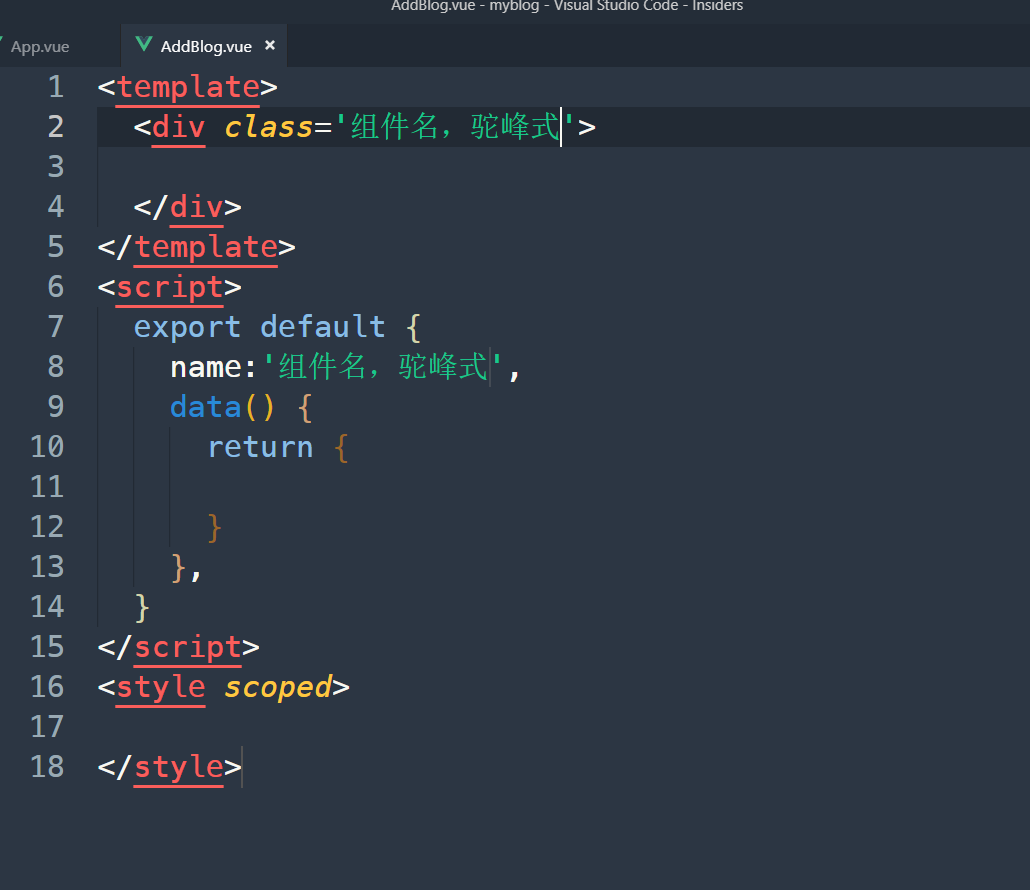




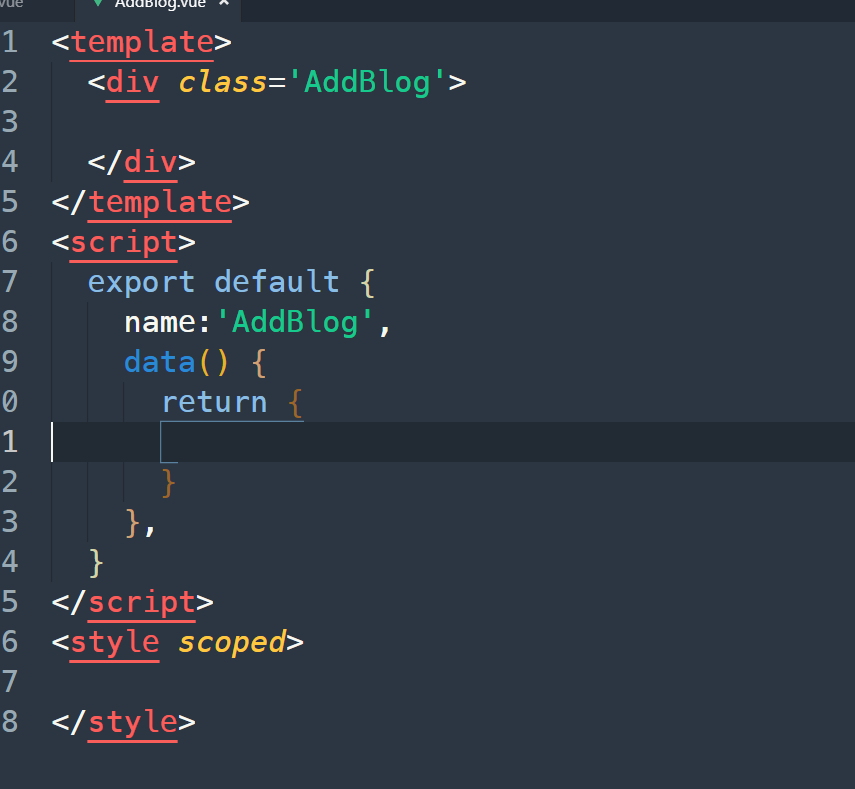
## 创建AddBlog组件

打开components文件夹

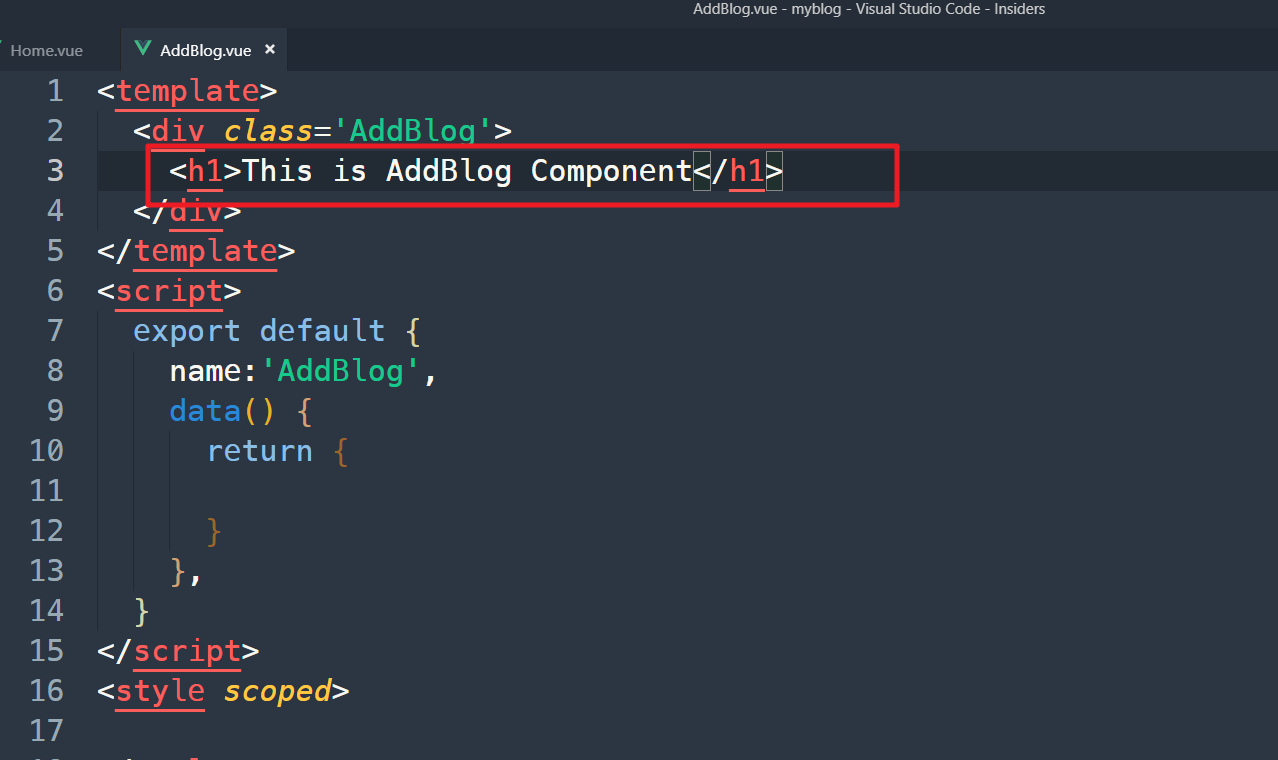
创建AddBlog.vue,输入vtpl,按下回车



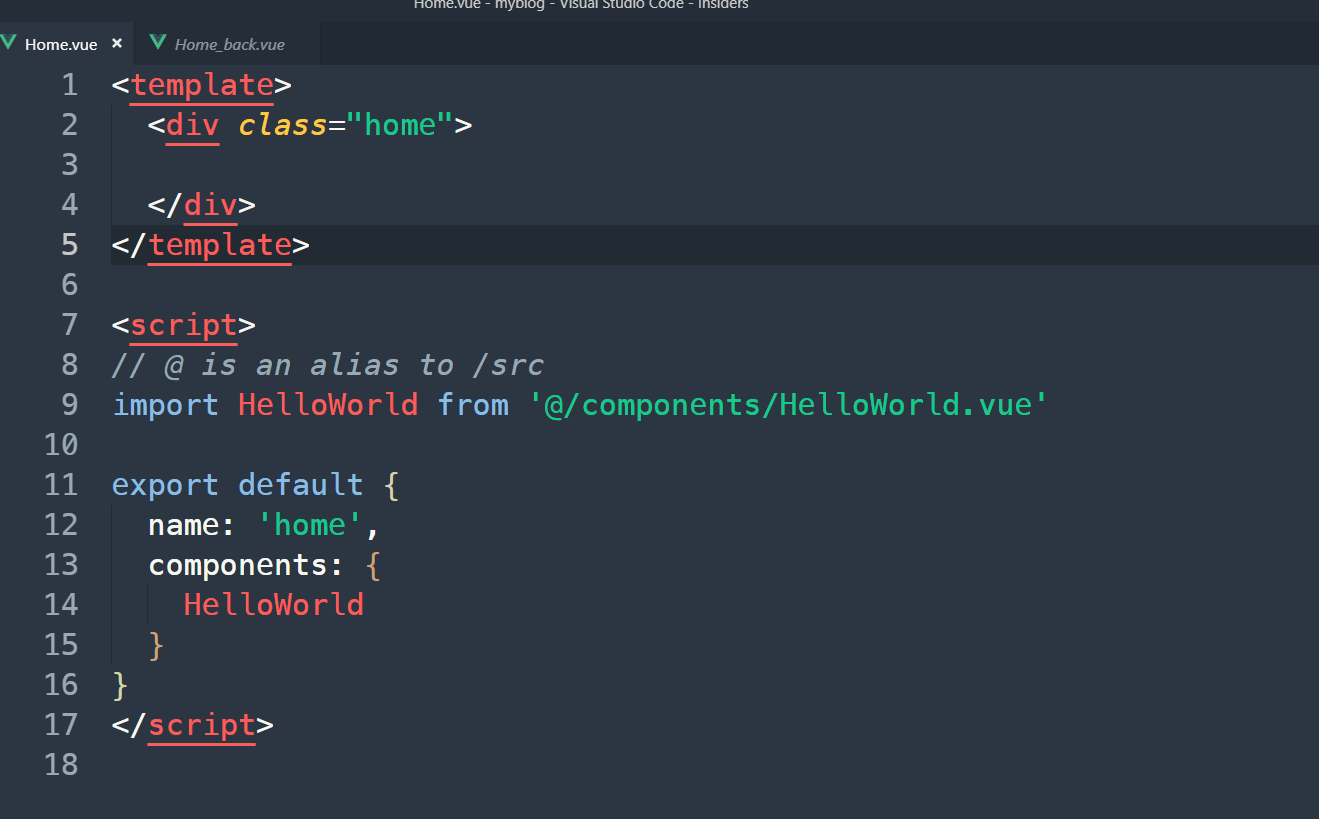
修改好名字



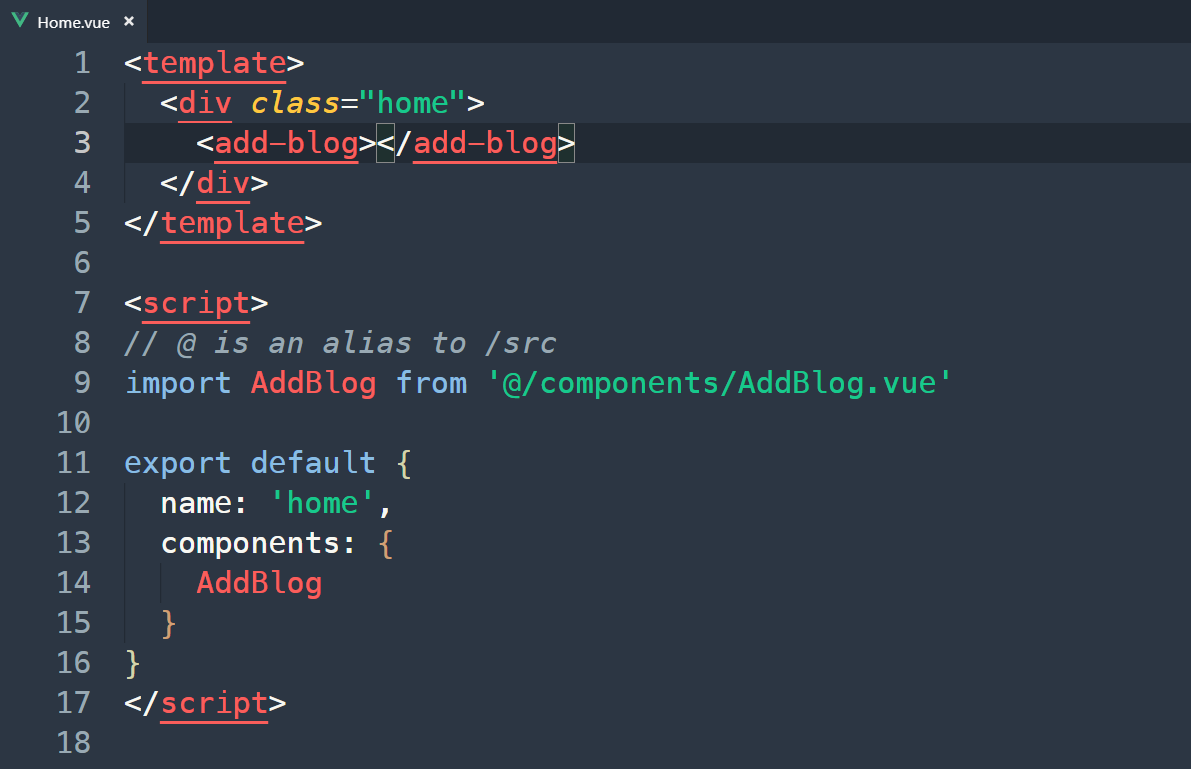
可以加一行代码：

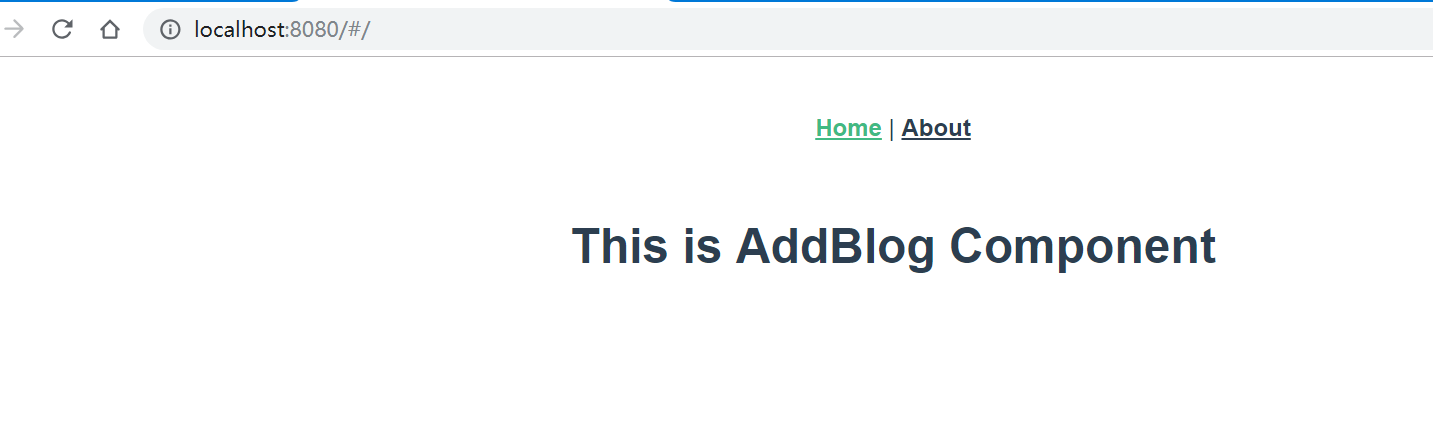


打开views/home.vue,然后把里面东修改如下：

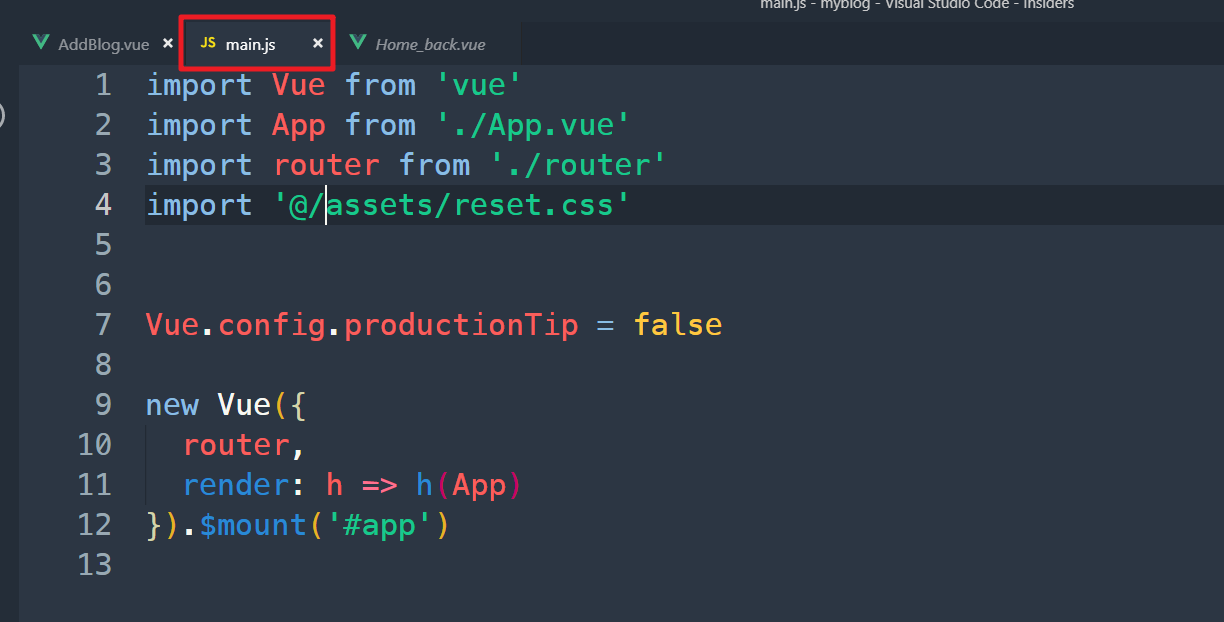


然后添加AddBlog组件





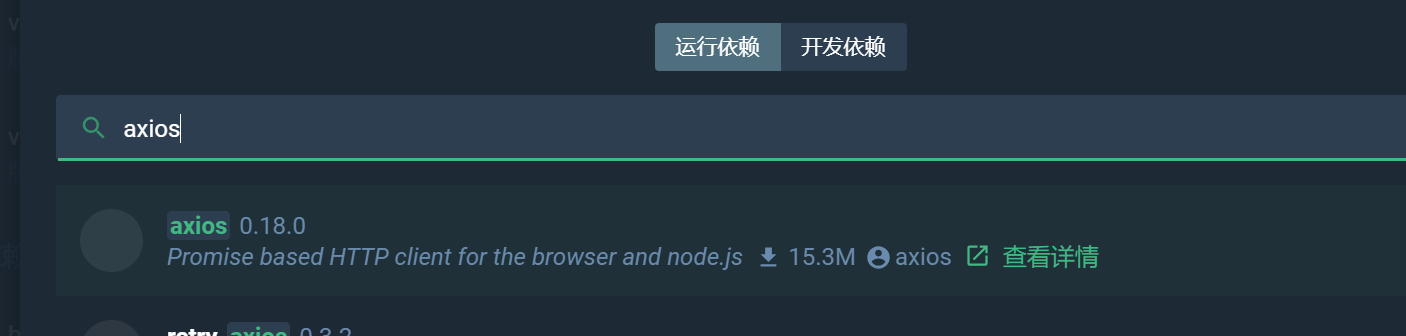
## 修改AddBlog组件的内容



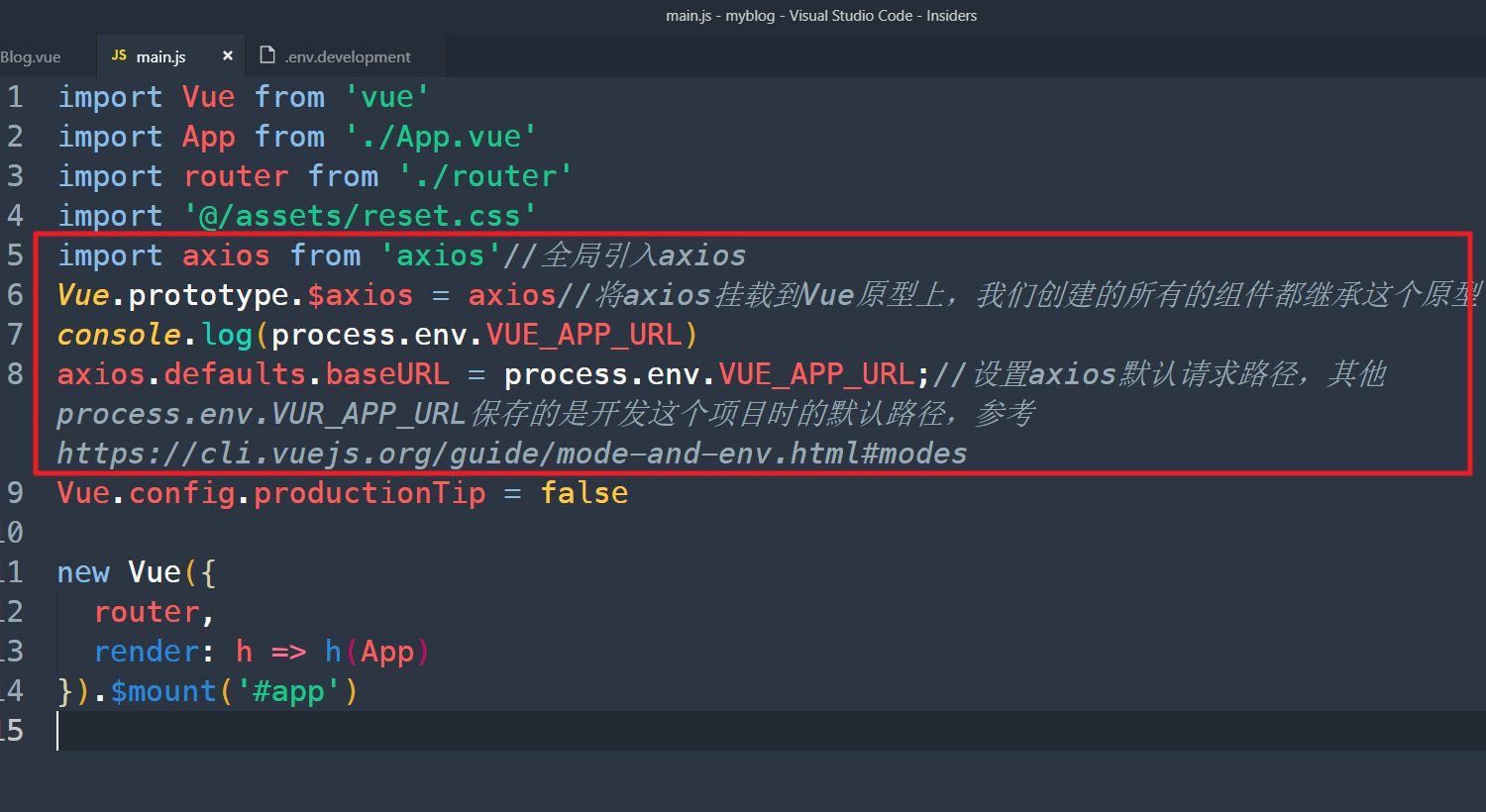
AddBlog.vue内容如下：

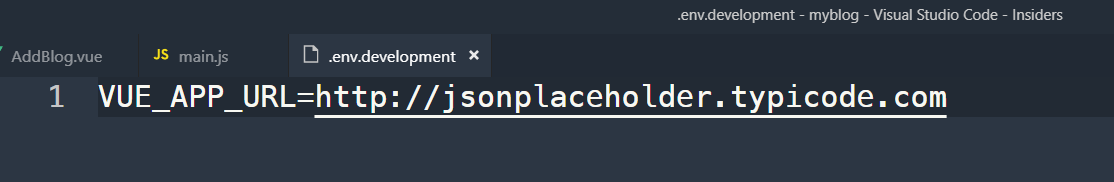
|  |
| --- |
| <template>  <div class='AddBlog'>  <h2>添加博客</h2>  <form>  <div>  <label for="title">博客标题</label>  <input type="text" name="" id="title" v-model="blog.title">  </div>  <div>  <label for="content">博客内容</label>  <textarea name="" id="content" cols="30" rows="10" v-model="blog.content"></textarea>  </div>  <div id="checkboxes">  <label for="vue">Vue.js</label>  <input type="checkbox" id="vue" value="vue.js" v-model="blog.categories">  <label for="node">Node.js</label>  <input type="checkbox" id="node" value="node.js" v-model="blog.categories">  <label for="react">React.js</label>  <input type="checkbox" id="react" value="React.js" v-model="blog.categories">  <label for="angular">Angular.js</label>  <input type="checkbox" id="angular" value="Angular.js" v-model="blog.categories">  </div>  <div>  <select name="" id="" v-model="blog.author">  <option v-for="(item,index) in authors" :key="index">  {{item}}  </option>  </select>  </div>  <button @click.prevent="post">添加博客</button>  </form>  <hr>  <div id="preview">  <h3>博客总览</h3>  <p>博客标题:{{blog.title}}</p>  <p>博客内容:{{blog.content}}</p>  <ul>  <li v-for="(item,index) in blog.categories" :key="index">  {{ item }}  </li>  </ul>  <p>作者:{{blog.author}}</p>  </div>  </div>  </template>  <script>  export default {  name:'AddBlog',  data() {  return {  blog:{  title:'',  content:'',  categories:[],  author:''  },  authors:['Hemiah','Henry','Bucky']  }  },  methods: {  post() {  console.log(111)  }  },  }  </script>  <style scoped>  </style> |

## 添加axios





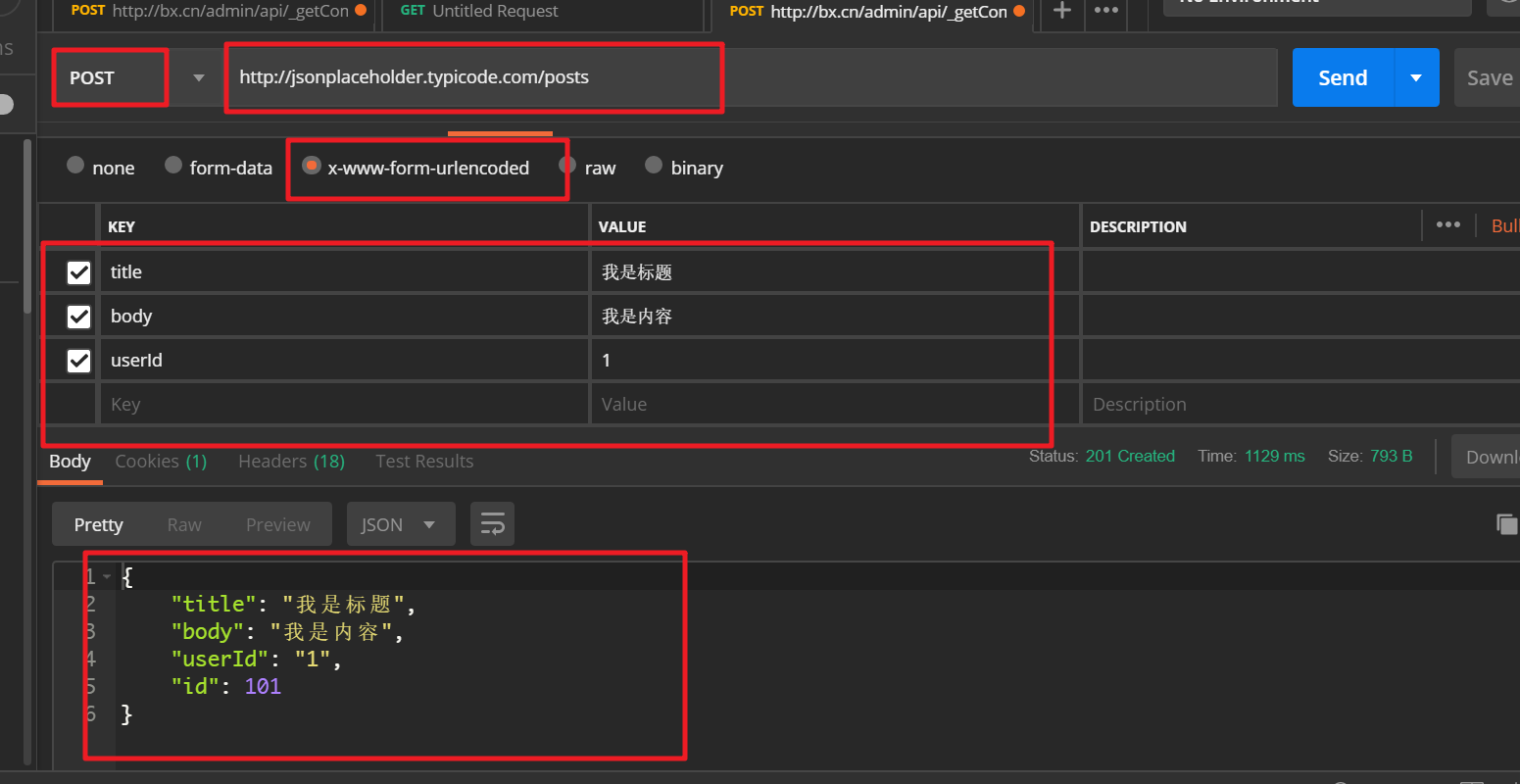




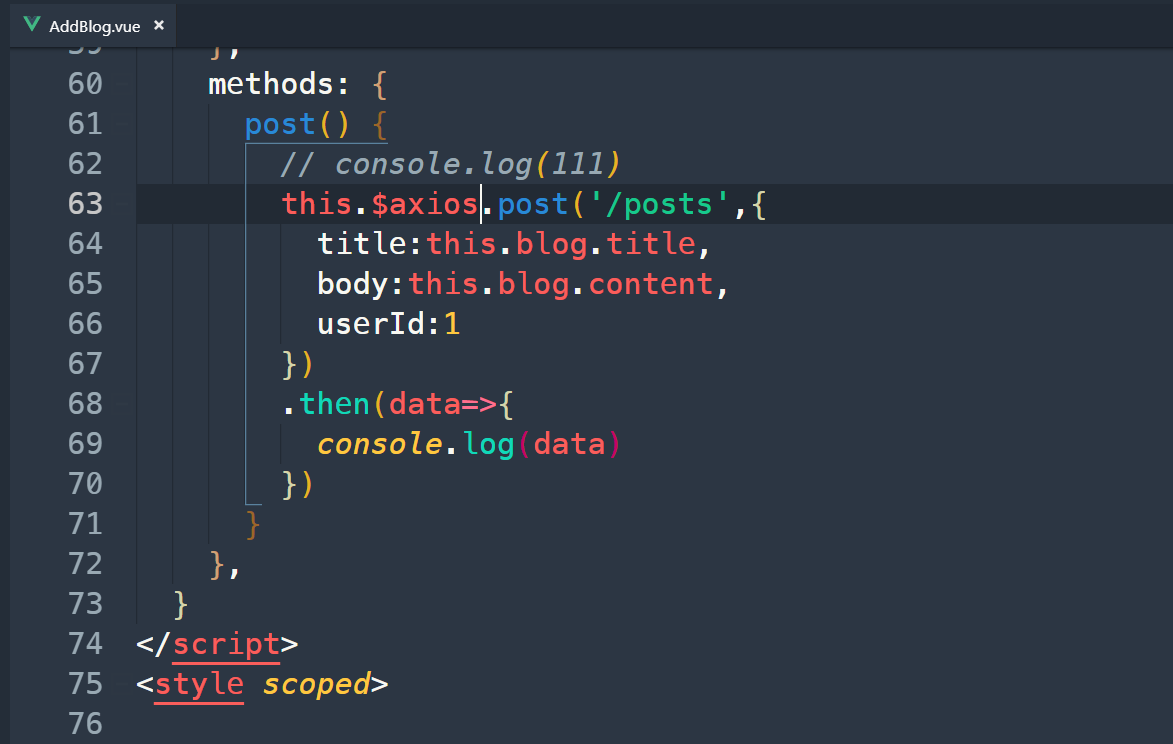
## 使用axios请求数据

<http://jsonplaceholder.typicode.com/>

首先，我们使用postman来测试数据



数据没有问题，我们开始写代码



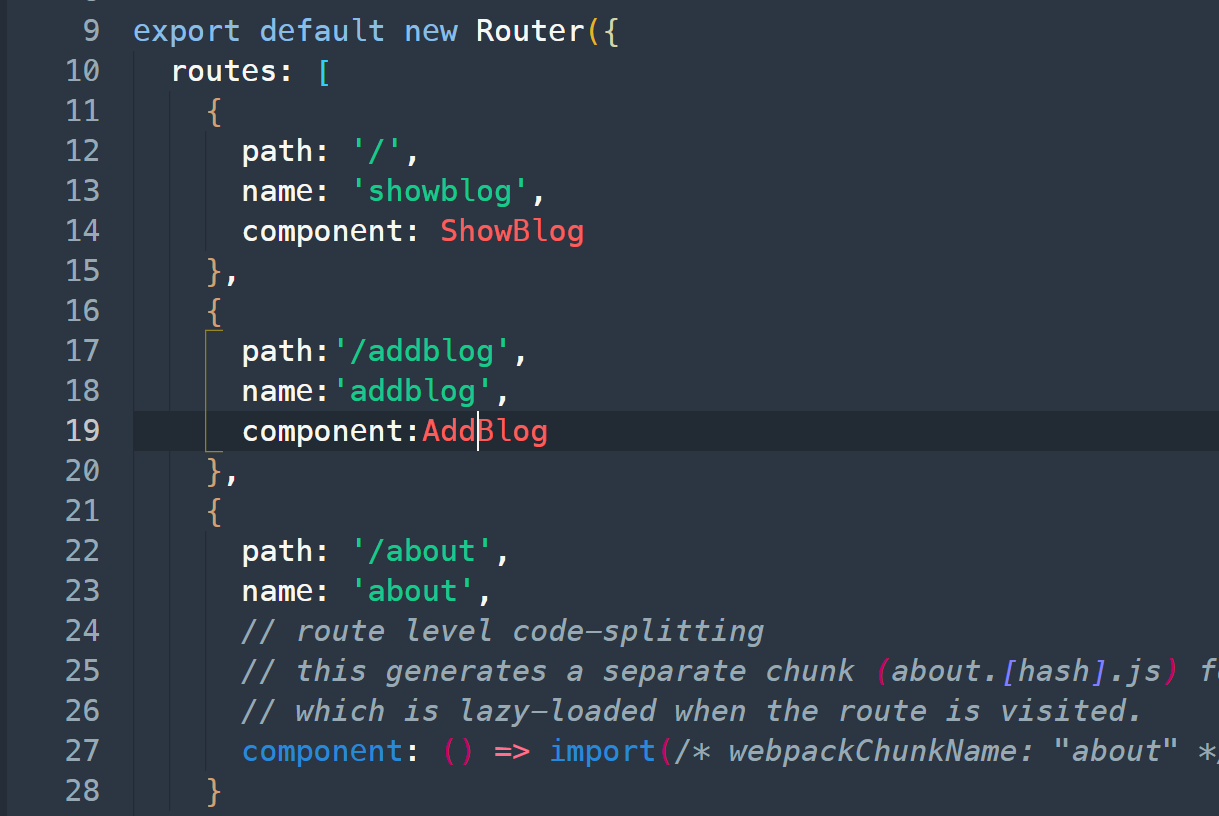
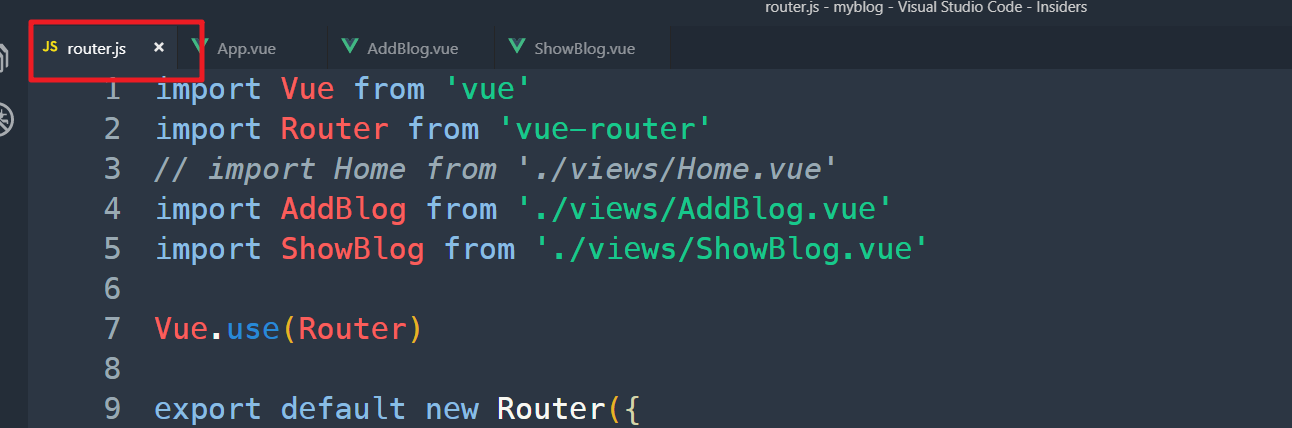
## 只显示博客总览内容

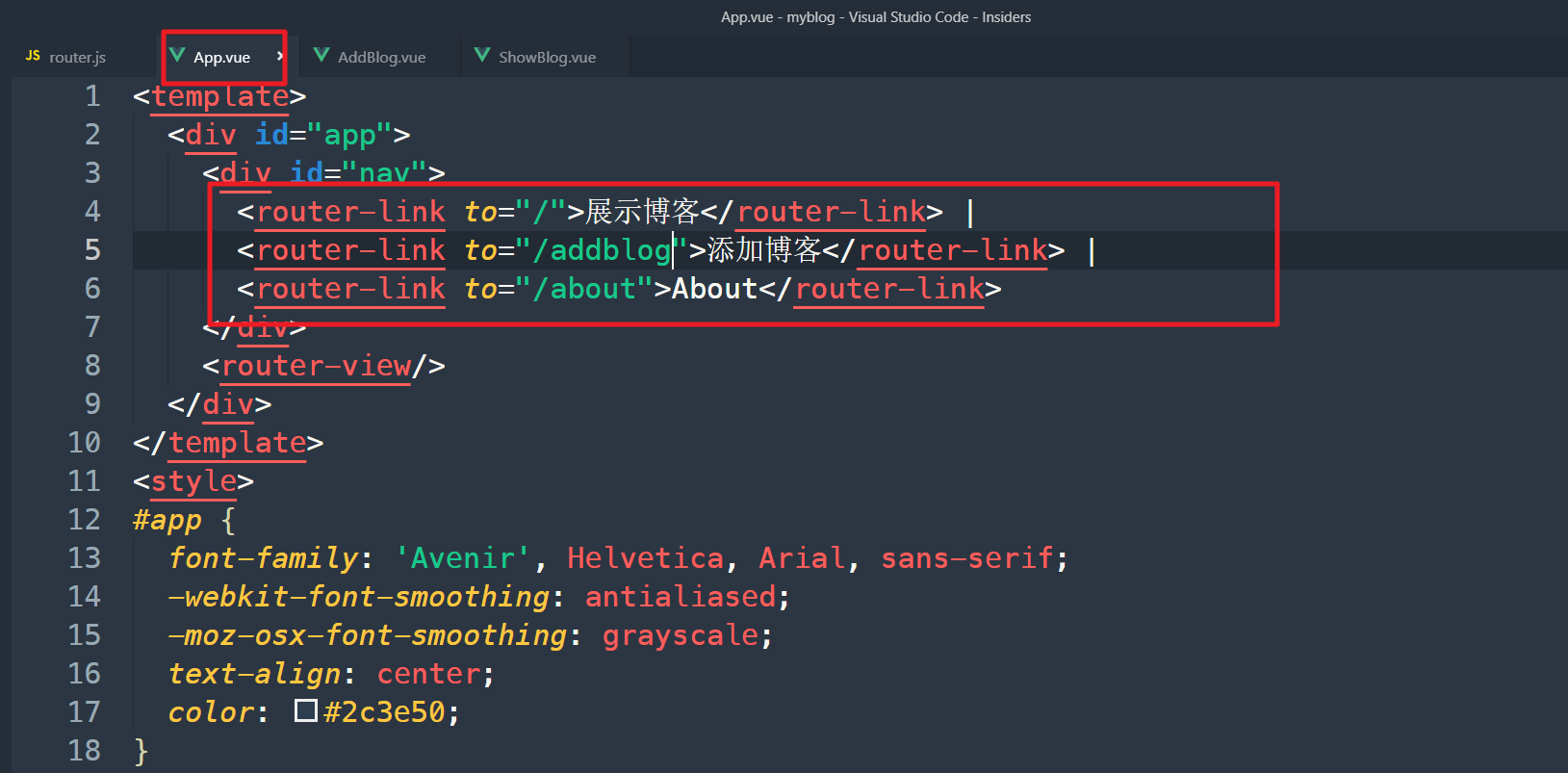
|  |
| --- |
| <template>  <div class='AddBlog'>  <h2>添加博客</h2>  <form v-if="!submmited">  <div>  <label for="title">博客标题</label>  <input type="text" name="" id="title" v-model="blog.title">  </div>  <div>  <label for="content">博客内容</label>  <textarea name="" id="content" cols="30" rows="10" v-model="blog.content"></textarea>  </div>  <div id="checkboxes">  <label for="vue">Vue.js</label>  <input type="checkbox" id="vue" value="vue.js" v-model="blog.categories">  <label for="node">Node.js</label>  <input type="checkbox" id="node" value="node.js" v-model="blog.categories">  <label for="react">React.js</label>  <input type="checkbox" id="react" value="React.js" v-model="blog.categories">  <label for="angular">Angular.js</label>  <input type="checkbox" id="angular" value="Angular.js" v-model="blog.categories">  </div>  <div>  <select name="" id="" v-model="blog.author">  <option v-for="(item,index) in authors" :key="index">  {{item}}  </option>  </select>  </div>  <button @click.prevent="post">添加博客</button>  </form>  <div v-if="submmited">  <h3>您的博客发布成功!!</h3>  </div>  <div id="preview">  <h3>博客总览</h3>  <p>博客标题:{{blog.title}}</p>  <p>博客内容:{{blog.content}}</p>  <ul>  <li v-for="(item,index) in blog.categories" :key="index">  {{ item }}  </li>  </ul>  <p>作者:{{blog.author}}</p>  </div>  </div>  </template>  <script>  export default {  name:'AddBlog',  data() {  return {  blog:{  title:'',  content:'',  categories:[],  author:''  },  authors:['Hemiah','Henry','Bucky'],  submmited:false  }  },  methods: {  post() {  // console.log(111)  this.$axios.post('/posts',{  title:this.blog.title,  body:this.blog.content,  userId:1  })  .then(data=>{  console.log(data)  this.submmited = true  })  }  },  }  </script>  <style scoped>  </style> |

## 路由规划

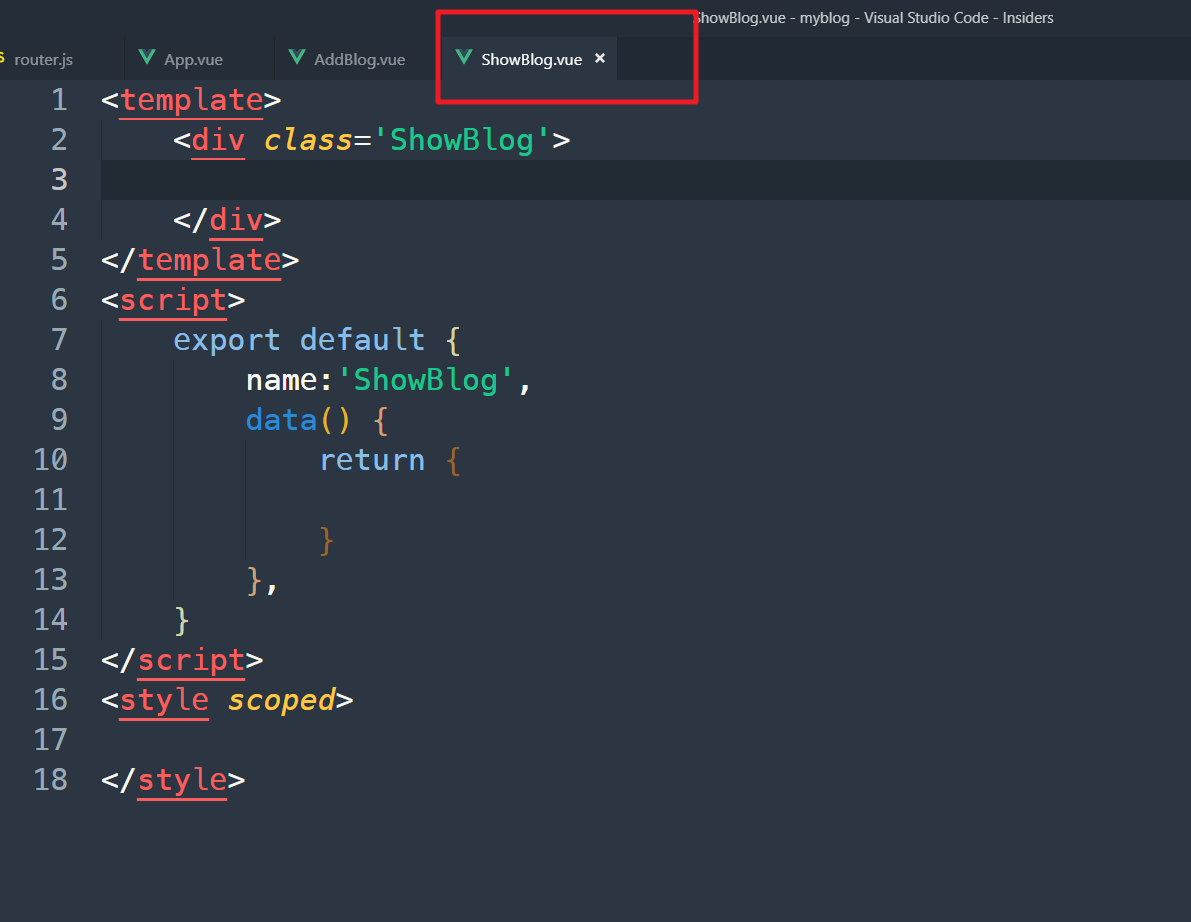
我们规划有两个路由：AddBlog和ShowBLog

views和router是一一对应的

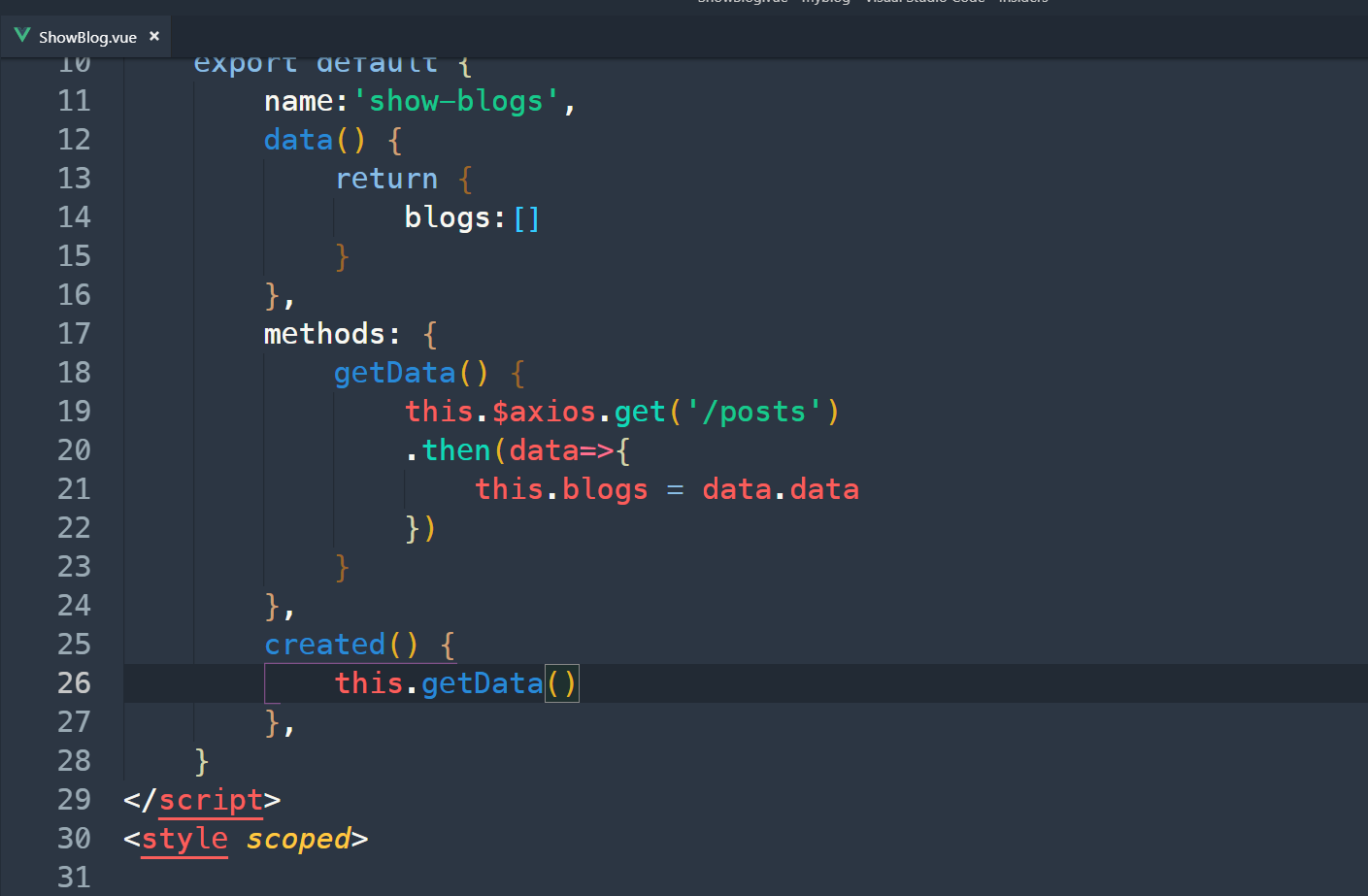


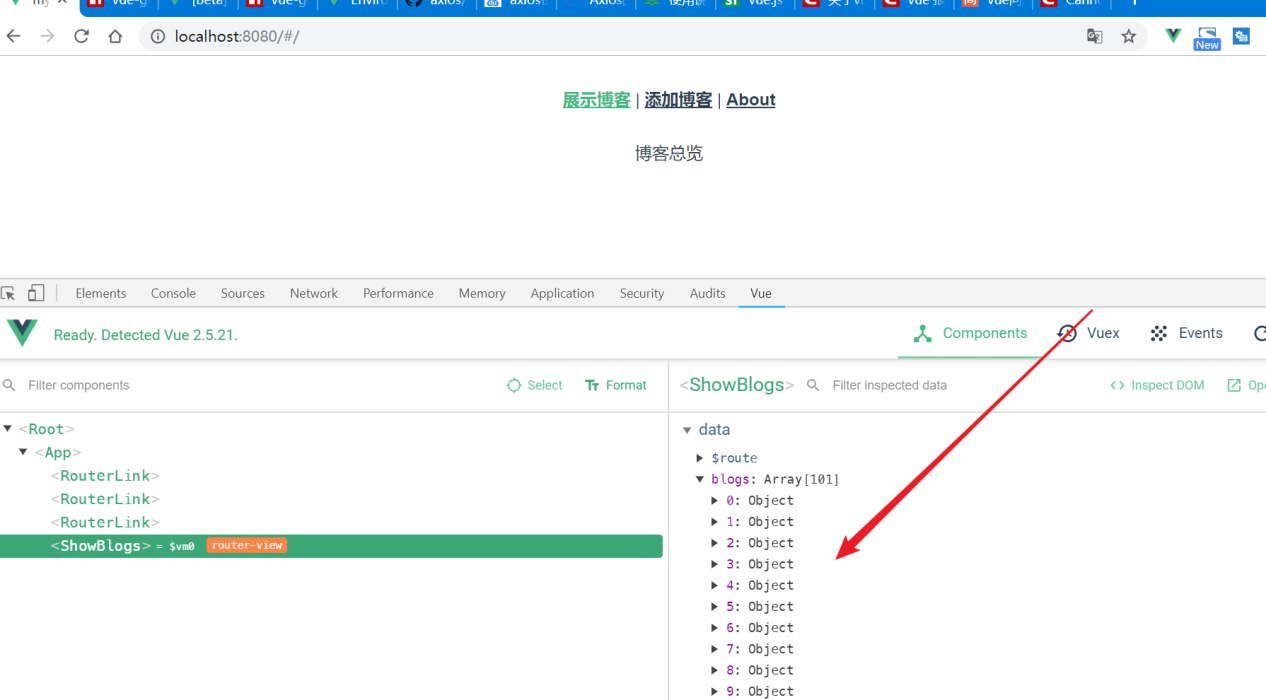


把components中的AddBLog.vue添加到views当中,在views中创建ShowBlog.vue

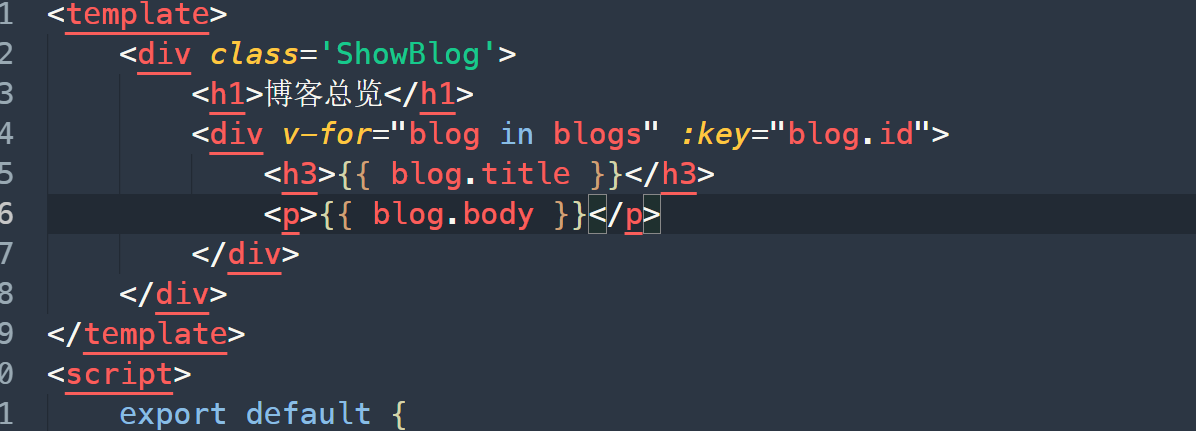


## 展示博客



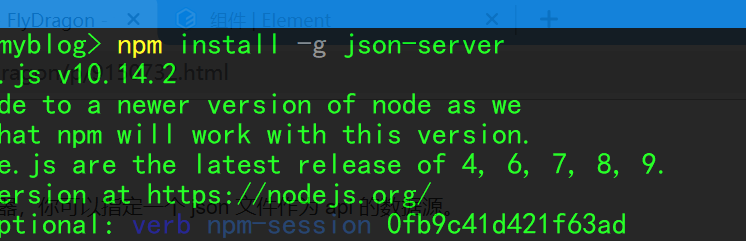




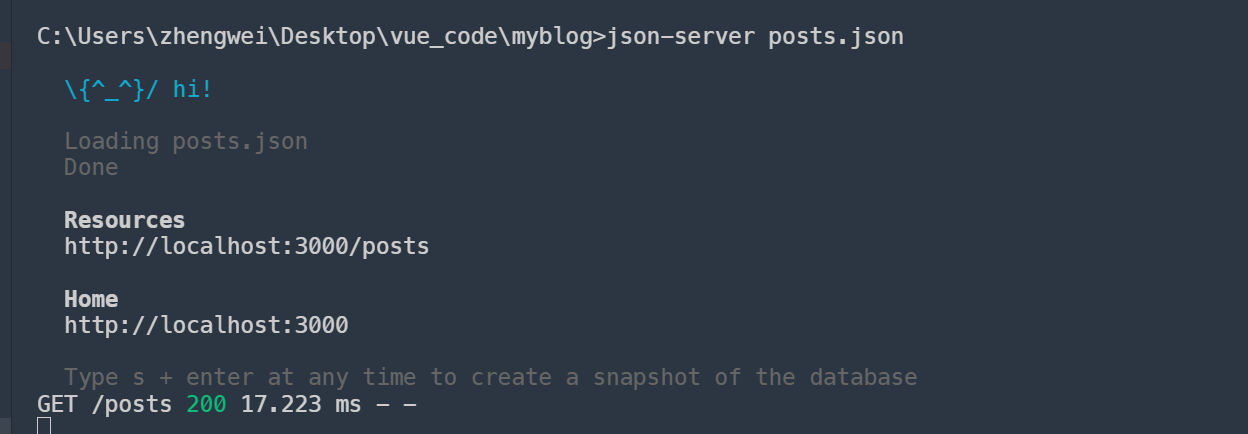


## 把线上的数据替换成本地的假数据

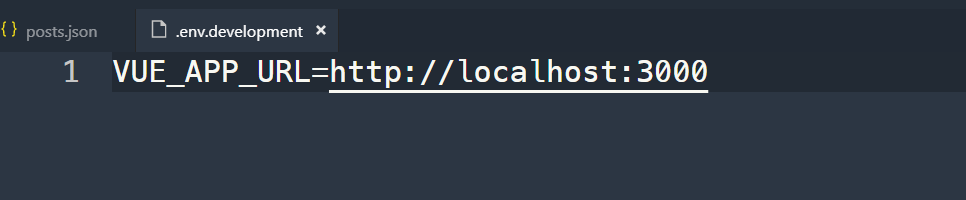
第一步、安装json-server



第二步、执行如下代码：



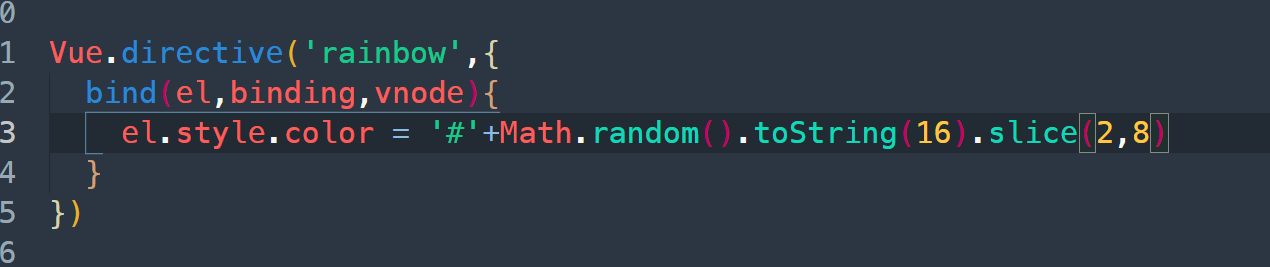
第三步、修改.env.development

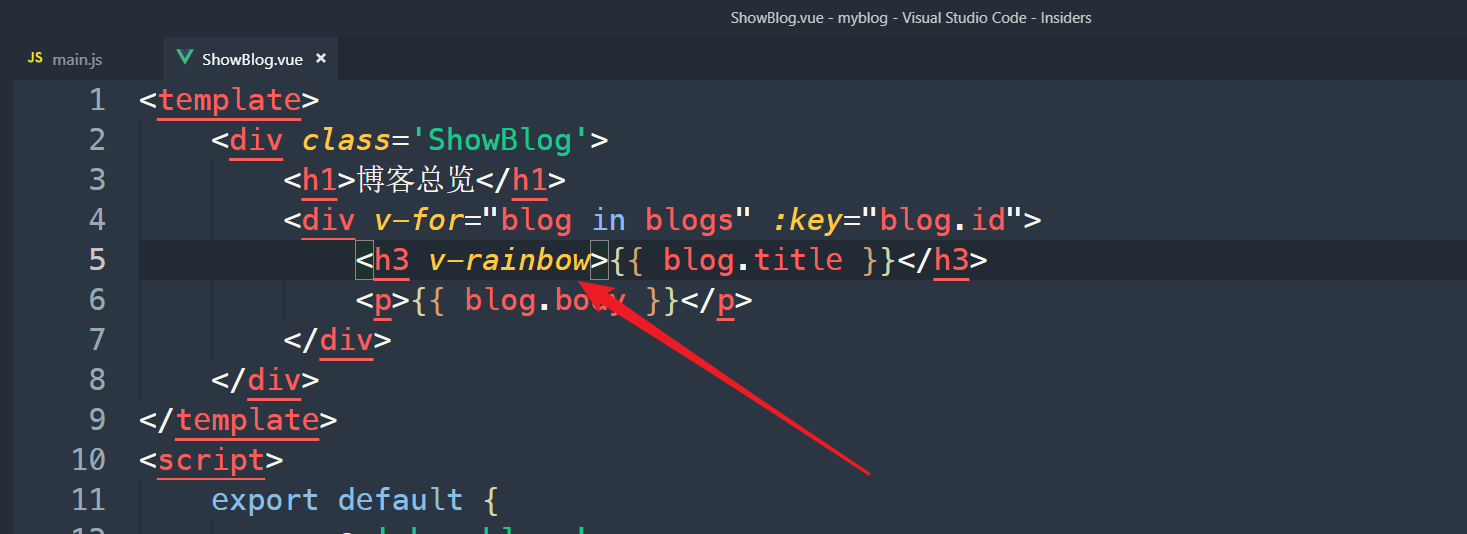


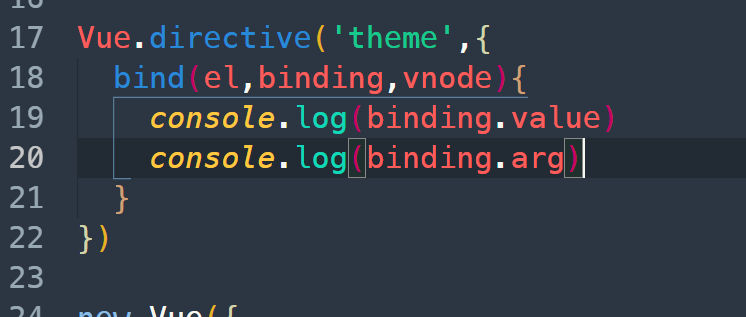
其他代码不要动

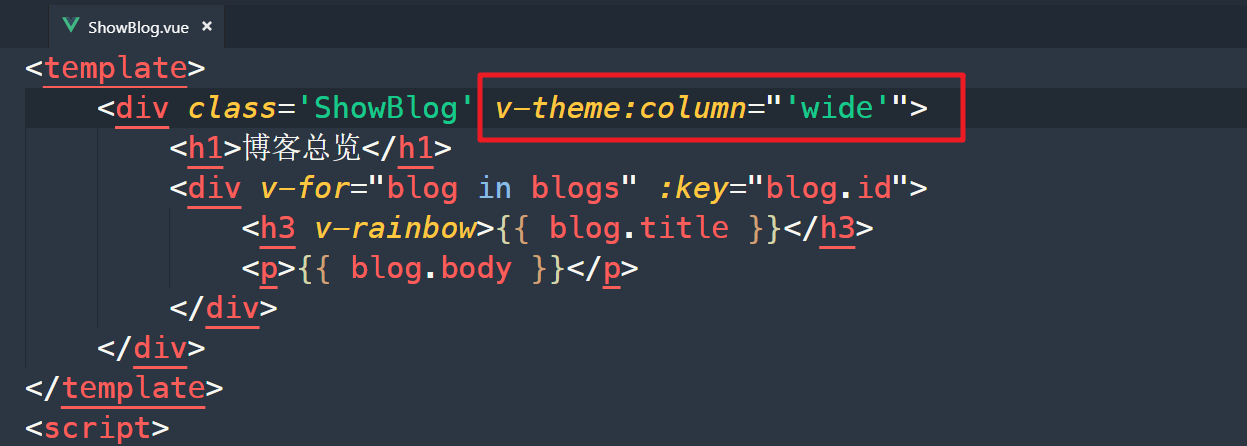
## 自定义指令

在main.js当中定义一个全局指令

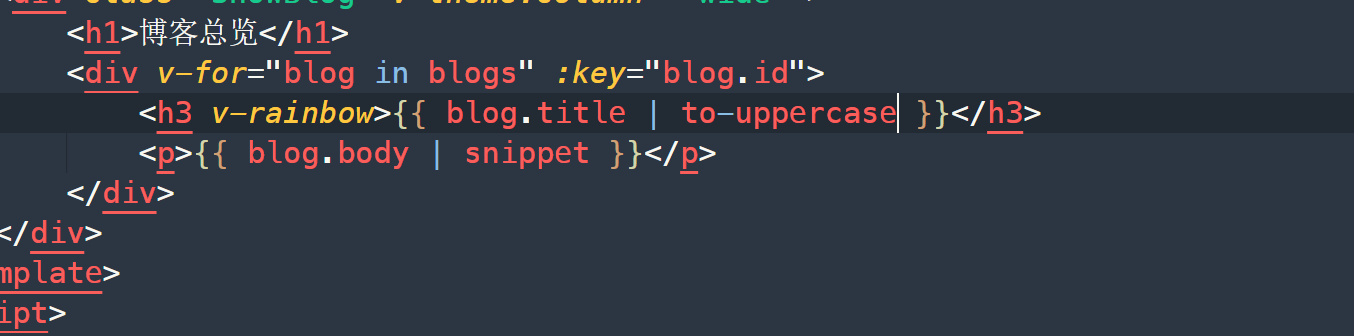


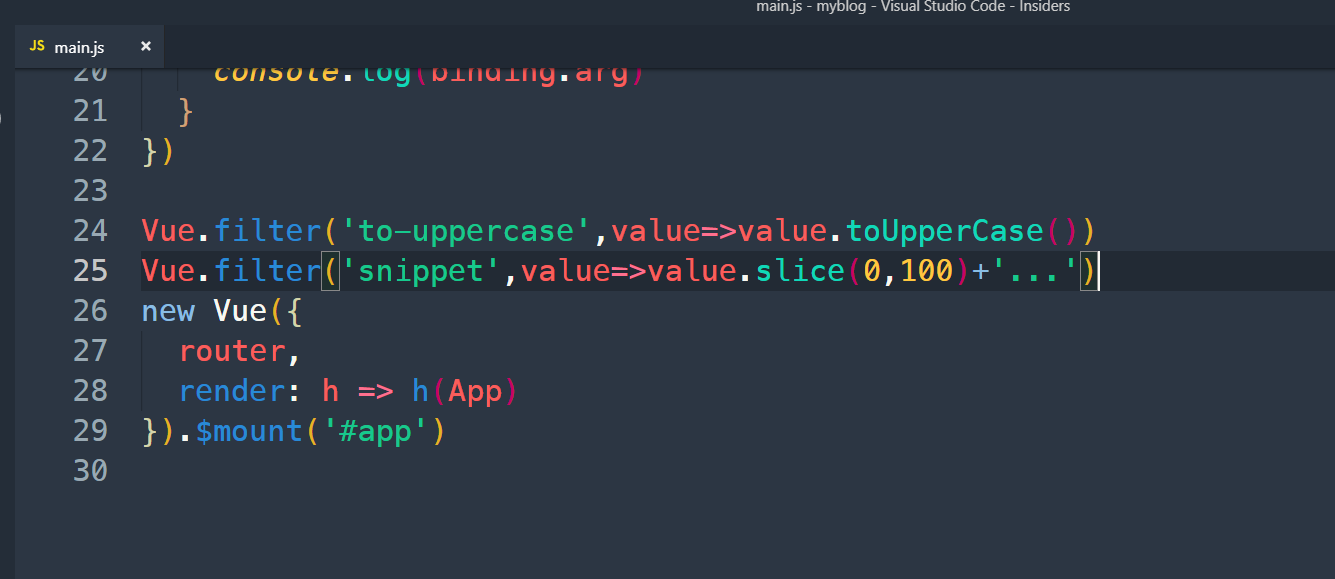






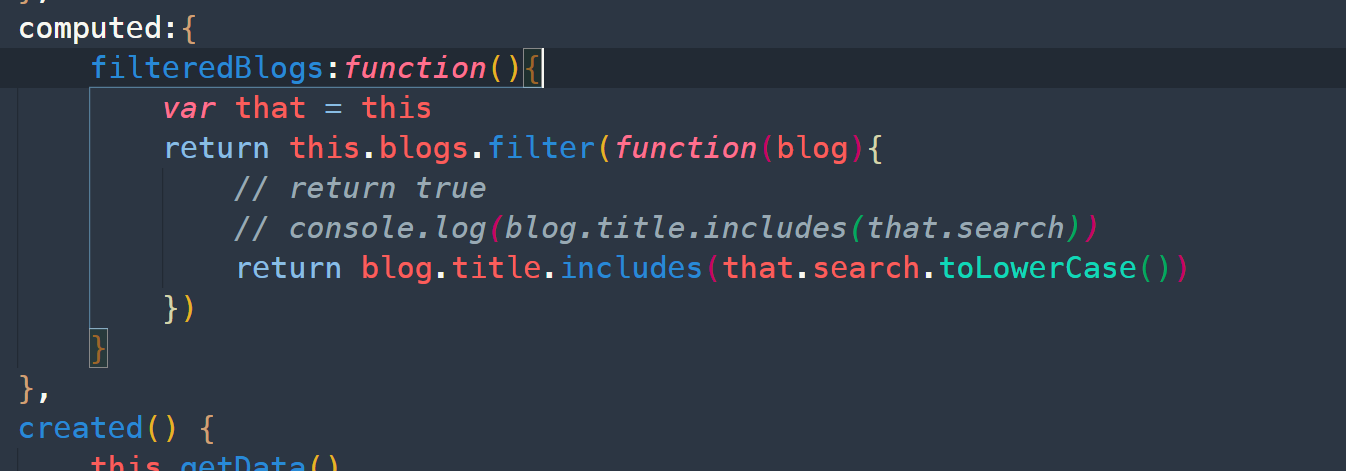
## 过滤器

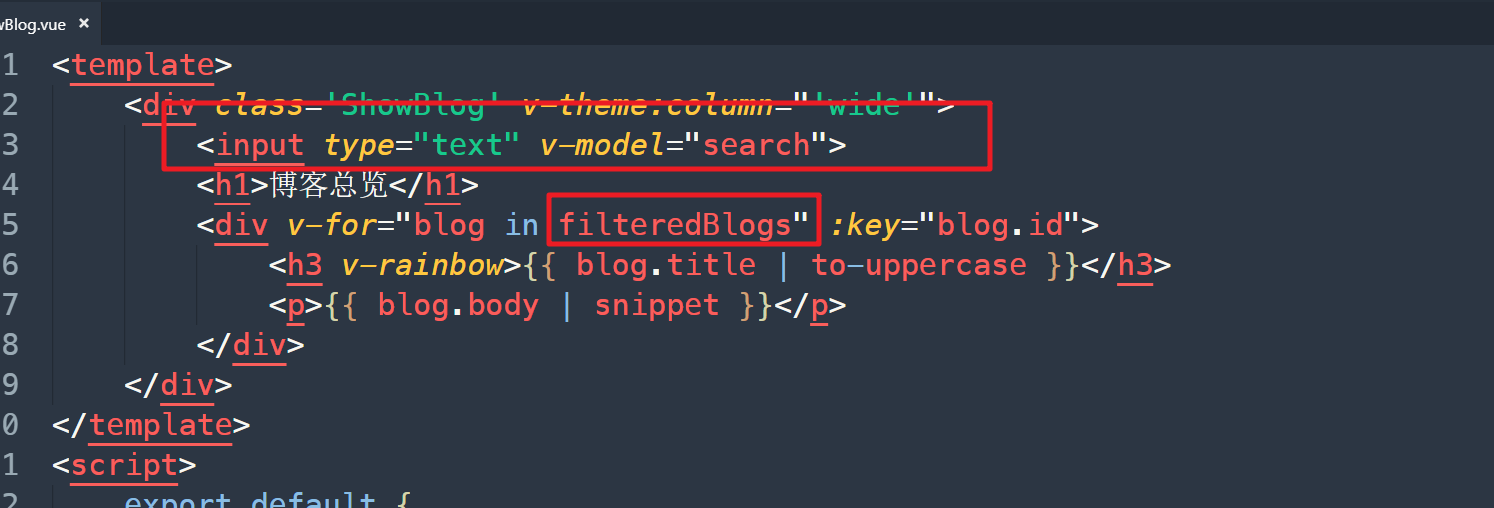




## 搜索功能

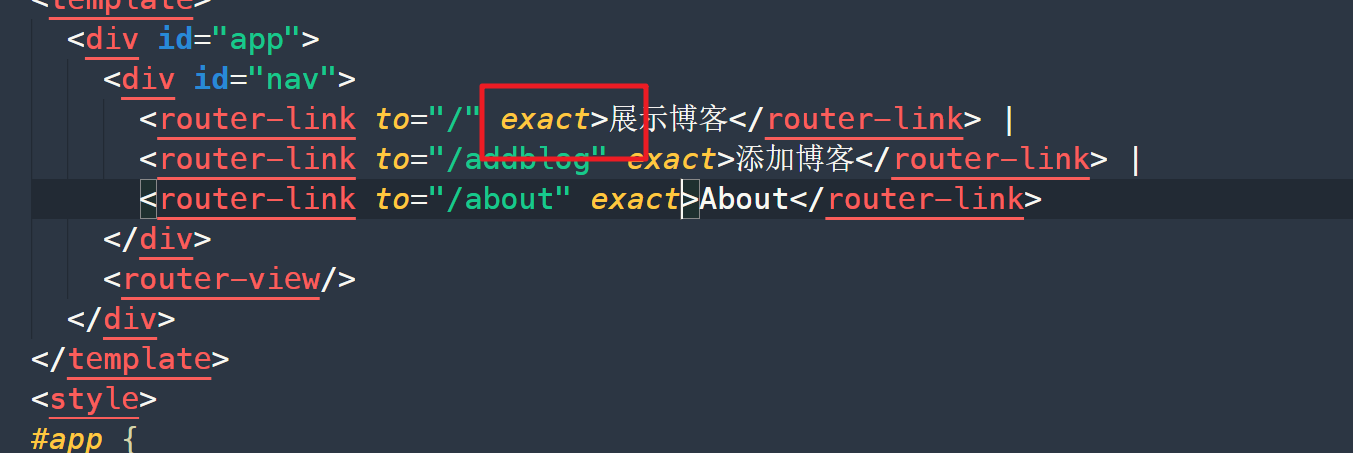
注意事项：对象的方法不能用箭头函数





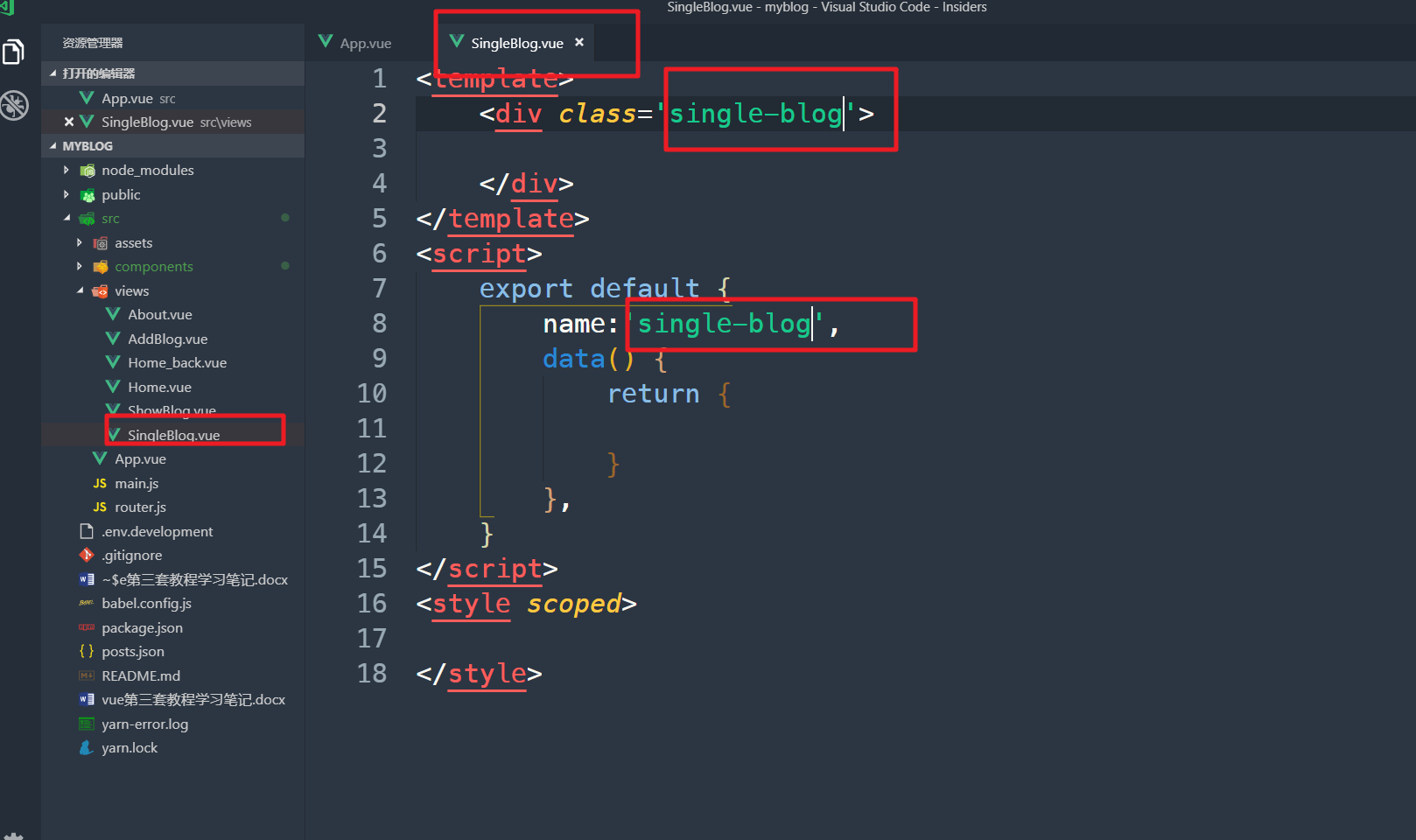
## 把自定义指令、过滤器修改为局部



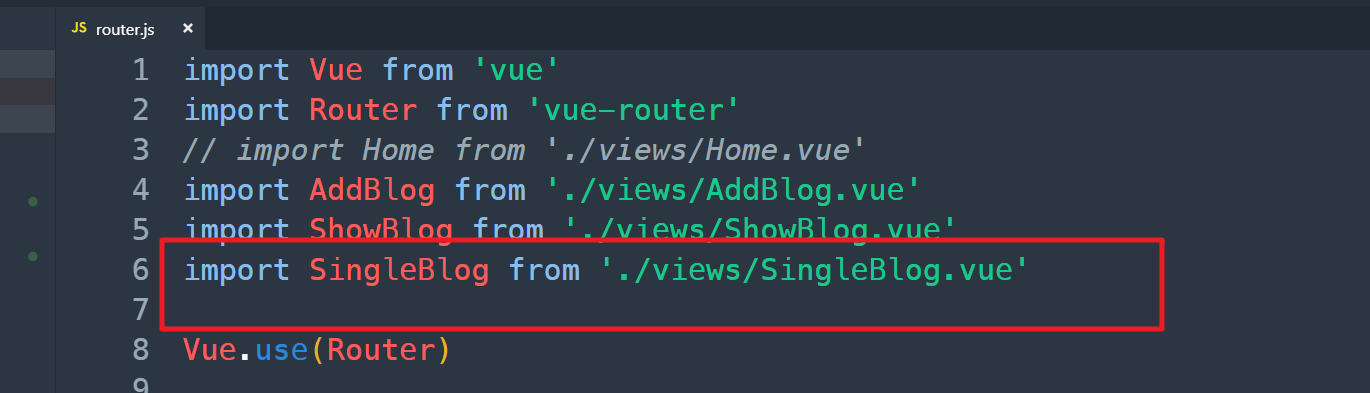


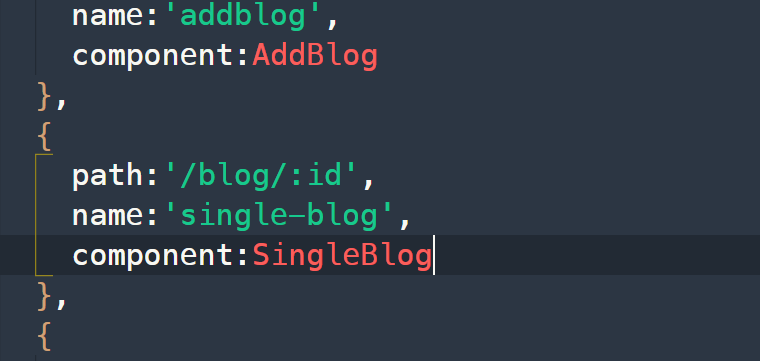
## 详情页制作

第一步、在views当中添加SingleBlog.vue

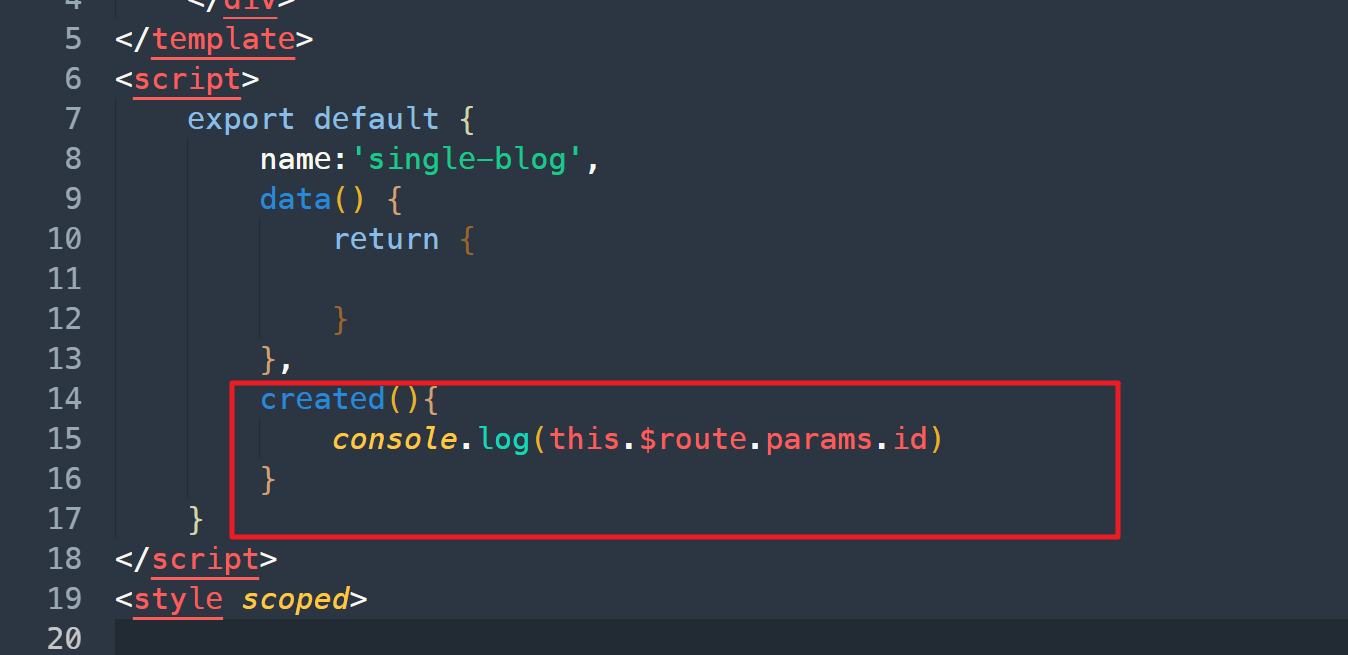


第二步、添加对应的路由





这块只有一个知识点，如何在vue当中获取url中的id值是多少

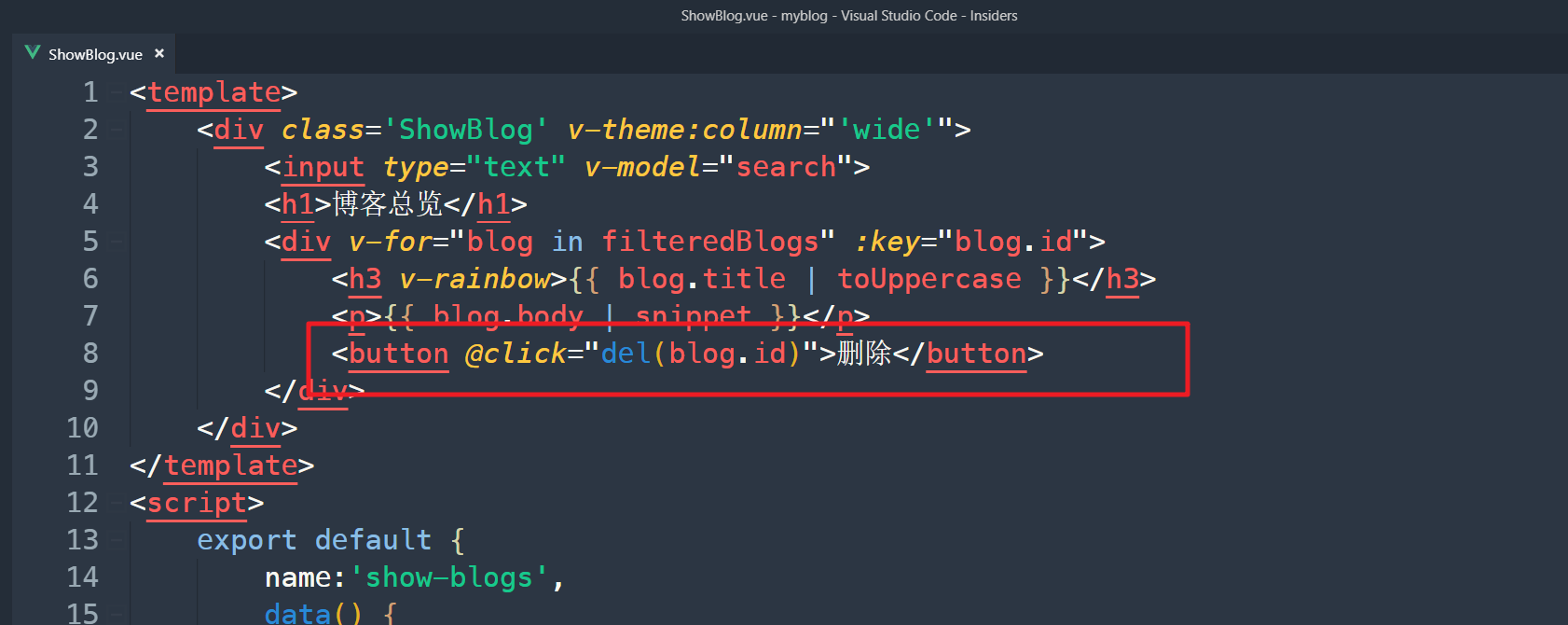


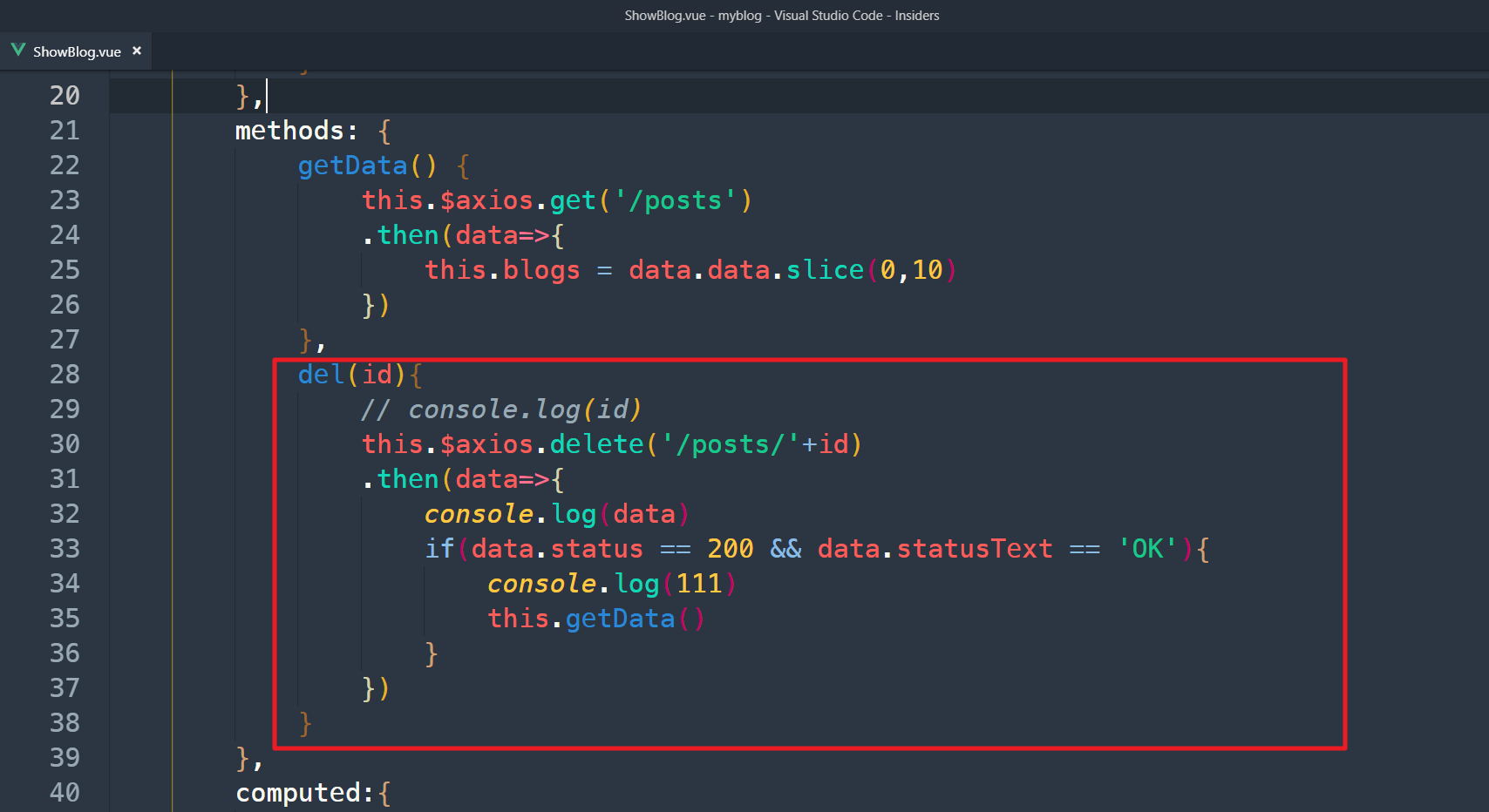
## firebase

包括后面的野狗之类的，没必要去看



## 删除功能





## 打包

