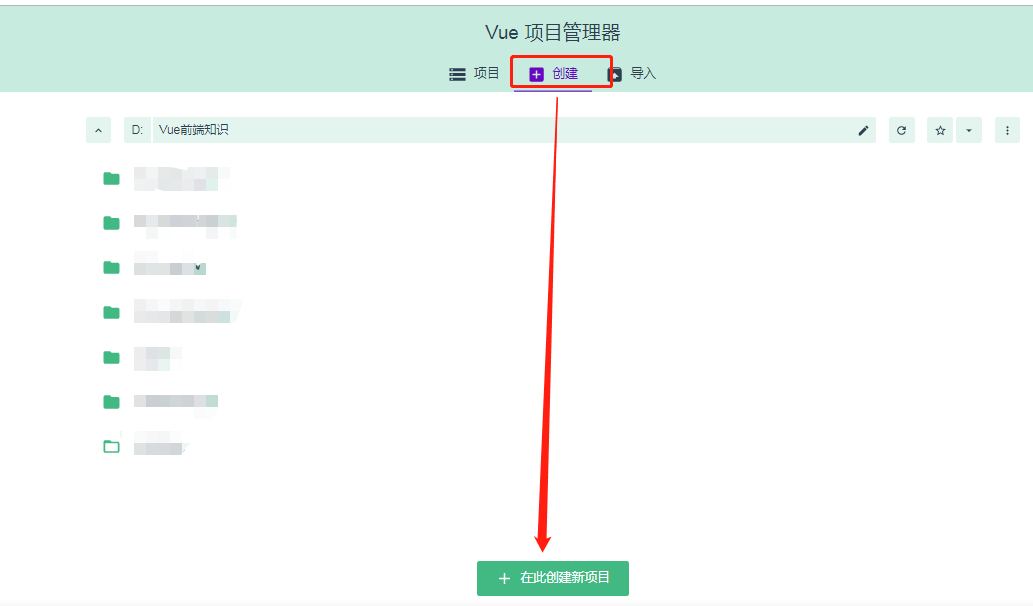
**Vue项目搭建步骤**

1. 创建项目

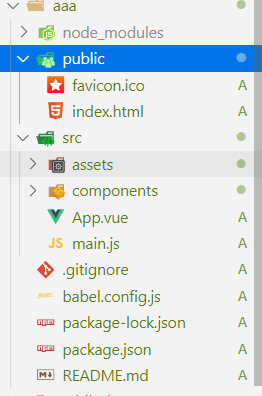
终端中启动vue ui



自动打开浏览器，创建项目



回到VS Code编辑器，可看到创建的项目，项目结构如下：



1. 配置项目端口

我们看到在创建项目是默认端口号为8080，和tomcat端口号重复，这里可以修改端口号。创建vue.config.js，内容如下：

module.exports = {

    devServer:{

      port:8083

    }

   };

1. 创建登录组件

在component文件夹下面创建组件Login，用于登录页面展示，内容如下：

<template>

  //页面代码展示

</template>

<script>

export default {

//数据绑定和方法

}

</script>

<style>

//页面css

</style>

复制代码和css如下：

|  |
| --- |
| 页面代码：  <div id="login">       <div id="content" >            <div class="login-wrap">                <div class="login-con">                    <h1>管理员登录</h1>                    <form id="login\_form">                      <div class="login-same">                        <dl>                          <dt>账号：</dt>                          <dd><input type="text" class="text" v-model="userName" value="" id="userName" placeholder="邮箱/手机/用户名"></dd>                        </dl>                         <dl>                          <dt>密码：</dt>                          <dd><input type="password" class="text" value="" v-model="password" placeholder="请输入您的密码"></dd>                        </dl>                         <dl>                            <dt>&nbsp;</dt>                          <dd>                          <input type="button" class="login\_ent" @click="login()" id="loginBtn" value="登录">                          </dd>                        </dl>                      </div>                    </form>                </div>            </div>       </div>    </div> |
| Css样式：  body,div,a,img,h1,h2,h3,h4,h4,h5,p,ol,ul,li,dl,dt,dd{ margin:0; padding:0; list-style:none; border:none;}  body{ font-size:12px; font-family:"宋体"; color:#616161; }  a{ color:#999; text-decoration:none;}  a:hover{ text-decoration:underline;}  /\*content部分\*/  #content{      width:100%;      height:800px;      background:url(../assets/img/bg.jpg) no-repeat 0 0;      margin:0;      overflow:hidden;  }  .login-wrap{      margin:200px auto;      width:800px;      overflow: hidden;  }  .login-con{      width:600px;      height: 300px;      border: 1px solid #d8d8d8;      border-radius: 5px;      background: #f9f9f9;      margin-left:50px;  }  .login-con h1{      color: #555;      font-family: "microsoft yahei";      font-size: 18px;      font-weight: 700;      line-height: 20px;      padding: 15px 0 25px 25px;      text-align: center;  }  .login-same{      overflow: hidden;      padding: 0 100px;  }  .login-same dl{      clear: both;      float: left;      padding: 0;      width: 100%;      margin:10px;  }  .login-same dl dt{      color: #777;      float: left;      font-size: 14px;      line-height: 28px;      text-align: right;      width: 22%;  }  .login-same dl dd {      float: right;      text-align: left;      width: 75%;      position: relative;  }  .login-same dl dd .text {      background-color: #fff;      border: 1px solid #bbb;      font-family: Tahoma;      height: 20px;      line-height: 20px;      padding: 3px !important;      width: 200px;  }  .login\_ent {      background:#f08300;      border: 0 none;      border-radius:2px;      color: #fff;      cursor: pointer;      font-size: 14px;      font-weight: 700;      height: 34px;      margin: 0;      padding: 0;      vertical-align: middle;      width: 210px;      float: left;      text-align: center;  }  div.load{      position: absolute;      left: 560px;      z-index: 9999;      top: 200px;      display:none;  } |
| 登录数据绑定：   data(){          return{             userName:'',             password:''          }      },      methods:{      } |

1. 使用vue-router路由实现转发:

安装vue-router：cnpm i vue-router –D

src下创建router.js

引入vue-router，创建VueRouter路由对象，并将路由对象暴露出去对象

|  |
| --- |
| import VueRouter from "vue-router"  var routerObj = new VueRouter({  routes: [    ]  });  //将路由对象暴露出去  export default routerObj; |

5、修改入口文件main.js，

注册并使用vue-router，

引入router.js，

并将路由对象挂载到vue上：

|  |
| --- |
| import Vue from 'vue'  import App from './App.vue'  import VueRouter from "vue-router"  import routerObj from './router.js'  Vue.use(VueRouter);  Vue.config.productionTip = false  new Vue({    render: h => h(App),  //将路由对象挂载到Vue    router:routerObj  }).$mount('#app') |

1. 修改App.vue，用于展示路由匹配到的组件内容：

|  |
| --- |
| <template>   <div id="app">     <router-view></router-view>    </div>  </template>  <script>  export default {    name:"app"  }  </script>  <style>  </style> |

1. router.js中定义路由转发，实现Login.vue页面展示：

在routes中配置如下：

|  |
| --- |
| {      //为/路径设定重定向，进入项目时，自动转至/login的连接地址      path:"/",      redirect:"/login"    },    {      path: "/login",      component: Login    } |

8、运行项目：npm run serve

显示如下：



**使用axios做前后端分离，实现页面登录**

1. Vue 框架开发的时候，会遇到跨域的问题，可在vue.config.js 里配置proxy 代理。

|  |
| --- |
| module.exports = {   devServer:{     port:8083,     proxy: {      '/MobileShop': {          target: 'http://127.0.0.1:8080/MobileShop',           changeOrigin: true,          ws: true,          pathRewrite: {              '^/MobileShop': '/'          }        }      }   }  }; |

2、引入axios 和 qs

cnpm i axios

cnpm i qs

main.js引入：

|  |
| --- |
| import axios from "axios"  import qs from "qs"  Vue.prototype.axios = axios  Vue.prototype.qs = qs |

3、点击登录，执行login()方法

|  |
| --- |
| login(){     this.axios.post('/MobileShop/admin\_manager/login',this.qs.stringify({'input': this.userName,password:this.password}))            .then(response=>{      if(response.data.status==0){  this.$router.push({path:"/main"});       }else{         alert(response.data.msg);       }     }).catch(function(error){          console.log(error);     })  } |

4、创建Main.vue，存放登录成功跳转页面。

|  |
| --- |
| <template>    <div>欢迎进入</div>  </template>  <script>  export default {  }  </script>  <style>  </style> |

5、登录成功，实现页面跳转：

Router.js中import引入Main.vue，

配置/main路由转发

|  |
| --- |
| import Main from "./components/Main" |
| {        path: "/main",        component: Main  } |