

# 第6章 网站前台-登陆与用户中心

学习目标：

- 完成用户注册功能
- 完成用户登陆功能，掌握js-cookie的使用
- 完成微信扫码登陆的功能
- 完成用户中心嵌套布局，掌握nuxt嵌套路由的使用

## 1 用户注册

### 1.1 页面构建

创建pages/login.vue



```
<template>
  <div class="wrapper loginsign">
    <div class="item signup">
      <div class="form">
        <h3 class="loginsign-title">注册新账号</h3>
        <form class="sui-form">
          <div class="control-group">
            <label for="inputname" class="control-label">名字</label>
            <div class="controls">
              <input type="text" id="inputname" placeholder="真实姓名或常用昵称"
class="input-xlarge" data-rules="required" />
            </div>
          </div>
          <div class="different">
            <div class="radio-content">
              <div id="a1" class="phone">
                <div class="control-group number">
                  <input type="text" placeholder="仅支持大陆手机号" class="input-
xlarge" data-rules="required|mobile" />
                </div>
                <div class="control-group code">
                  <div class="input-append">
                    <input id="appendedInputButton" type="text" placeholder="短信验
证" class="span2 input-large msg-input" />
                    <button type="button" class="sui-btn msg-btn">获取验证
码</button>
                  </div>
                </div>
              </div>
            </div>
            <div class="control-group">
              <label for="inputpassword" class="control-label">密码</label>
              <div class="controls">
                <input type="text" id="inputpassword" placeholder="请输入6-16位
密码" class="input-xlarge" />
              </div>
            </div>
            <div id="a2" class="email">
              <div class="control-group inputemail">

                <input type="text" placeholder="输入手机号" class="input-xlarge"

```



```

/>
    </div>
    <div class="control-group">
        <label for="inputpassword" class="control-label">密码: </label>
        <div class="controls">
            <input type="text" id="inputpassword" placeholder="请输入6-16位
字符" class="input-xlarge" />
        </div>
    </div>
</div>
<div class="control-group btn-signup">
    <label class="control-label"></label>
    <div class="controls">
        <label> <input type="checkbox" /><span class="type-text"
style="font-size:12px;">同意协议并接受《服务条款》 </span> </label>
        <button type="submit" class="sui-btn btn-danger btn-yes">注 册
</button>
    </div>
</div>
</form>
</div>
<div class="item">
    <div class="form">
        <h3 class="loginsign-title">用户登录</h3>
        <form class="sui-form login-form">
            <div class="control-group">
                <label for="inputname" class="control-label">手机号或
Email: </label>
                <div class="controls">
                    <input type="text" id="inputname" placeholder="11位手机号或Email"
class="input-xlarge" data-rules="required" />
                </div>
            </div>
            <div class="control-group">
                <label for="inputpassword" class="control-label">密码: </label>
                <div class="controls">
                    <input type="text" id="inputpassword" placeholder="输入登录密码"

```

```
class="input-xlarge" />
    </div>
  </div>
  <div class="controls">
    <label> <input type="checkbox" name="remember-me" /><span
class="type-text" style="font-size:12px;">记住登录状态</span> </label>
    <button type="submit" class="sui-btn btn-danger btn-yes">登
录</button>
  </div>
  <div class="other-methods">
  </div>
</form>
</div>
</div>
</div>
</template>
<script>
import '~/assets/css/page-sj-person-loginsign.css'
export default {

}
</script>
```

其中的内容参见person-loginsign.html

## 1.2 获取验证码

### 1.2.1 模拟数据与API

- (1) 将user.yml 导入easymock
- (2) 修改easy-mock 数据

url: /user/user/sendsms/{mobile}

method: put



```
{  
  "code": 20000,  
  "flag": true,  
  "message": "验证码发送成功"  
}
```

### (3) 编写API 创建api/user.js

```
import request from '@/utils/request'  
const api_group = 'user'  
const api_name = 'user'  
export default {  
  sendsms(mobile) {  
    return request({  
      url: `/${api_group}/${api_name}/sendsms/${mobile}`,  
      method: 'put'  
    })  
  }  
}
```

## 1.2.2 调用API

### (1) 修改pages/login.vue脚本部分



```
<script>
import '~/assets/css/page-sj-person-loginsign.css'
import userApi from '@api/user'
export default {
  data(){
    return {
      pojo: {}
    }
  },
  methods: {
    sendsms(){
      userApi.sendsms( this.pojo.mobile ).then(res => {
        alert(res.data.message)
      })
    }
  }
}
</script>
```

(2) 修改pages/login.vue 绑定变量

```
<input type="text" v-model="pojo.mobile" placeholder="仅支持大陆手机号"
class="input-xlarge" />
```

(3) 修改pages/login.vue 按钮绑定方法

```
<button type="button" class="sui-btn msg-btn" @click="sendsms" >获取验证
码</button>
```

## 1.2.3 使用ElementUI 的弹出框

(1) 安装element-ui

```
cnpm install element-ui --save
```

(2) plugins文件夹下创建element-ui.js

```
import Vue from 'vue'
import ElementUI from 'element-ui'
Vue.use(ElementUI)
```

(3) 修改nuxt.config.js，加入插件与样式

```
.....
plugins: [
  ....
  { src: '~plugins/element-ui.js', ssr: false }
],
css: [
  'element-ui/lib/theme-chalk/index.css'
],
.....
```

(4) 修改pages/login.vue的脚本部分，将alert替换为以下代码

```
this.$message({
  message: res.data.message,
  type: (res.data.flag? 'success': 'error')
})
```

## 1.3 提交注册

(1) 在easy-mock 增加数据

URL: /user/user/register/{code}

Method: post

```
{
  "flag": true,
  "code": 20000,
  'message': "执行成功"
}
```

(2) 修改api/user.js，增加方法



```
register(user,code) {  
  return request({  
    url: `/${api_group}/${api_name}/register/${code}`,  
    method: 'post',  
    data:user  
  })  
},
```

(3) 修改pages/login/index.vue脚本部分 增加属性

```
data(){  
  return {  
    pojo: {},  
    code:''  
  }  
},
```

新增注册的方法

```
register () {  
  userApi.register(this.pojo).then( res =>{  
    if(res.data.flag){  
      this.$message({  
        message: '注册成功',  
        type: 'success'  
      })  
      this.pojo={}  
    }else{  
      this.$message({  
        message: '注册出错',  
        type: 'error'  
      })  
    }  
  })  
}
```

(4) 修改pages/login/index.vue页面部分

绑定表单输入框





```
<form class="sui-form">
  <div class="control-group">
    <label for="inputname" class="control-label">登录名</label>
    <div class="controls">
      <input type="text" id="inputname" v-model="pojo.loginname"
placeholder="登录名" class="input-xlarge" />
    </div>
  </div>
  <div class="control-group">
    <label for="inputname" class="control-label">昵称</label>
    <div class="controls">
      <input type="text" id="inputname" v-model="pojo.nickname"
placeholder="真实姓名或常用昵称" class="input-xlarge" />
    </div>
  </div>
  <div class="different">
    <div class="radio-content">
      <div id="a1" class="phone">
        <div class="control-group number">
          <input type="text" v-model="pojo.mobile" placeholder="仅支持大陆
手机号" class="input-xlarge" data-rules="required|mobile" />
        </div>
        <div class="control-group code">
          <div class="input-append">
            <input id="appendedInputButton" v-model="code" type="text"
placeholder="短信验证" class="span2 input-large msg-input" />
            <button type="button" class="sui-btn msg-btn" @click="sendsms"
>获取验证码</button>
          </div>
        </div>
      </div>
    </div>
    <div class="control-group">
      <label for="inputpassword" class="control-label">密码</label>
      <div class="controls">
        <input type="text" id="inputpassword" v-model="pojo.password"
placeholder="请输入6-16位密码" class="input-xlarge" />
      </div>
    </div>
  </div>
</div>
```

```
<div class="control-group btn-signup">
  <label class="control-label"></label>
  <div class="controls">
    <label> <input type="checkbox" /><span class="type-text"
style="font-size:12px;">同意协议并接受《服务条款》</span> </label>
    <button type="button" class="sui-btn btn-danger btn-yes"
@click="register">注 册</button>
  </div>
</div>
</form>
```

绑定方法

```
<button type="button" class="sui-btn btn-danger btn-yes"
@click="register">注 册</button>
```

## 1.4 输入校验

以下功能由学员实现：

- (1) 校验昵称必须填写
- (2) 校验手机号的合法性 ,可以使用正则表达式进行验证。

```
^((13[0-9])|(15[^4])|(18[0,2,3,5-9])|(17[0-8])|(147))\d{8}$
```

- (3) 密码长度校验
- (4) 判断是否勾选同意条款

## 2 用户登陆

### 2.1 登陆验证

- (1) mock模拟数据

url: /user/user/login

method: post



```
{
  "code": 20000,
  "flag": true,
  "data": {
    "token":
"eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI5ODcyNTUxMDQ1NDM1MjY5MTIiLCJpYXQiOiE1MjQyMTQ5NDgsInJvbnGVzIjoidXNlciIsImV4cCI6MTUyNDIxNTMwOH0.icFRMKfaHlPn224hU3Gm_LOHflaONj9IfWIVj8gSbbM",
    "name": "小白",
    "avatar": 'https://wpimg.wallstcn.com/f778738c-e4f8-4870-b634-56703b4acafe.gif'
  }
}
```

(2) API编写。 修改api/user.js, 新增方法

```
login(mobile, password) {
  return request({
    url: `/${api_group}/${api_name}/login`,
    method: 'post',
    data: {
      mobile,
      password
    }
  })
}
```

(3) 修改pages/login/index.vue , 增加属性: 用户名和密码

```
data(){
  return {
    ....
    mobile: '',
    password: ''
  }
},
```

(4) 修改pages/login/index.vue ,增加登陆的方法



```
login () {  
    userApi.login(this.mobile,this.password).then( res =>{  
        if(res.data.flag){  
            location.href='/manager'  
        }else{  
            this.$message({  
                message: res.data.message,  
                type: 'error'  
            })  
            this.mobile=''  
            this.password=''  
        }  
    })  
}
```

## (5) 绑定页面

```
<form class="sui-form login-form">
  <div class="control-group">
    <label for="inputname" class="control-label">手机号: </label>
    <div class="controls">
      <input type="text" id="inputname" v-model="loginname"
placeholder="11位手机号" class="input-xlarge" />
    </div>
  </div>
  <div class="control-group">
    <label for="inputpassword" class="control-label">密码: </label>
    <div class="controls">
      <input type="password" id="inputpassword" v-model="password"
placeholder="输入登录密码" class="input-xlarge" />
    </div>
  </div>
  <div class="controls">
    <label> <input type="checkbox" name="remember-me" /><span
class="type-text" style="font-size:12px;">记住登录状态</span> </label>
    <button type="button" @click="login" class="sui-btn btn-danger
btn-yes">登 录</button>
  </div>
  <div class="other-methods">
  </div>
</form>
```

测试效果，登陆后跳转到首页

## 2.2 登录用户信息存储

### (1) 安装js-cookie

```
cnpm install js-cookie --save
```

### (2) 创建utils/auth.js



```
import Cookies from 'js-cookie'

const TokenKey = 'User-Token'
const NameKey = 'User-Name'
const AvatarKey = 'User-Avatar'

export function setUser(token,name,avatar) {
  Cookies.set(NameKey, name)
  Cookies.set(AvatarKey, avatar)
  Cookies.set(TokenKey, token)
}
```

(3) 修改pages/login/index.vue 导入auth.js

```
import { setUser } from '@/utils/auth'
```

修改登陆方法，调用auth实现cookie的数据的保存

```
login(){
  userApi.login(this.mobile,this.password ).then( res=> {
    if(res.data.flag){
      //保存用户信息
      setUser(res.data.data.token, res.data.data.name,
res.data.data.avatar)
      location.href='/manager' //用户中心
    }else{
      this.$message( {
        message: res.data.message,
        type: "error"
      })
      this.mobile=''
      this.password=''
    }
  })
}
```

## 2.3 登陆状态显示用户信息

修改utils/auth.js

```
export function getUser() {  
  return {  
    token: Cookies.get(TokenKey),  
    name: Cookies.get(NameKey),  
    avatar: Cookies.get(AvatarKey)  
  }  
}
```

修改layouts/default.vue 代码部分

```
import { getUser } from '@/utils/auth'  
export default {  
  data() {  
    return {  
      user: {}  
    }  
  },  
  created() {  
    this.user = getUser()  
  }  
}
```

判断在当前已经登陆的情况下显示当前登录用户名称和头像

```
<div class="sui-nav pull-right info" v-if="user.name!==undefined">  
  <li><a href="/assets/other-notice.html" class="notice">{{user.name}}</a>  
</li>  
  <li><a href="#" class="homego"></a></li>  
</div>
```

## 2.4 未登录状态显示登陆链接

修改layouts/default.vue 页面部分

```
<div class="sui-nav pull-right info" v-if="user.name===undefined">  
  <router-link to="/login">登陆</router-link>  
</div>
```

## 2.5 退出登录

修改utils/auth.js

```
export function removeUser() {  
  Cookies.remove(TokenKey)  
  Cookies.remove(NameKey)  
  Cookies.remove(AvatarKey)  
}
```

修改layouts/default.vue 导入removeUser方法

```
import { getUser,removeUser } from '@/utils/auth'  
import userApi from '@/api/user'
```

增加退出登录的方法

```
methods:{  
  logout(){  
    removeUser()//清除登陆用户信息  
    location.href='/'  
  }  
}
```

增加退出登录的链接

```
<li><a @click="logout" class="notice">退出登录</a></li>
```

## 3 微信扫码登陆

### 3.1 账户申请

(1) 打开微信开放平台: <https://open.weixin.qq.com/> 首先进行账号的注册





## (2) 开发者资质认证

### (3) 创建网站应用



填写应用的相关信息



应用通过审核后，会得到AppID和 AppSecret,后边进行编码中会使用。

AppID : wx3bdb1192c22883f3

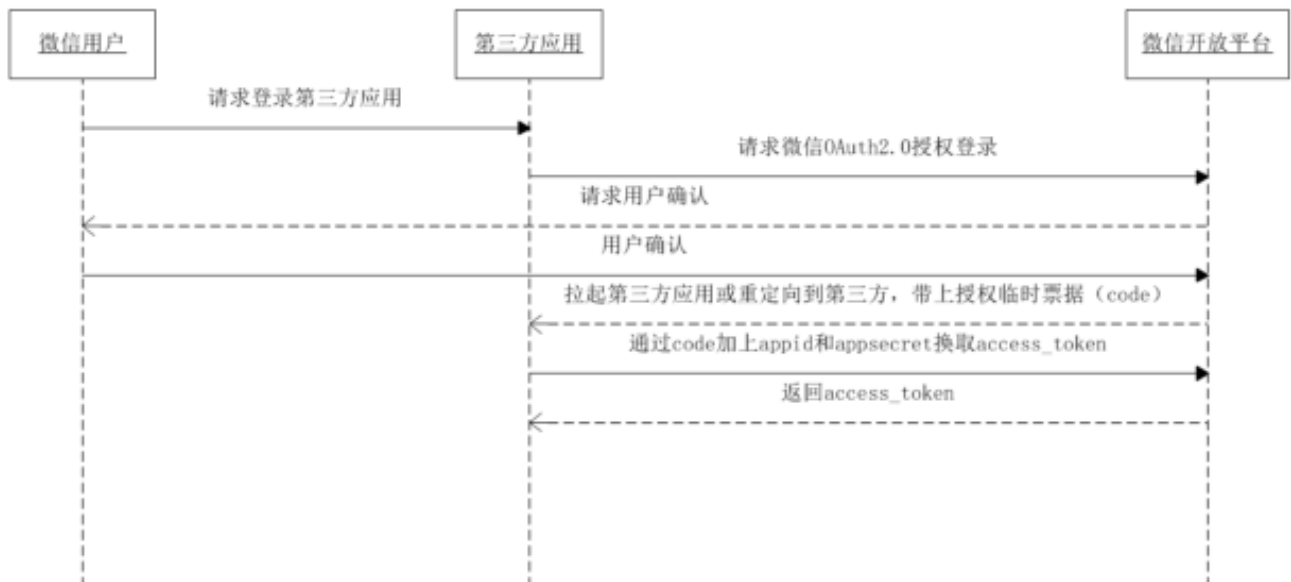
AppSecret : db9d6b88821df403e5ff11742e799105

## 3.2 微信第三方登陆流程

1. 第三方发起微信授权登录请求，微信用户允许授权第三方应用后，微信会拉起应用或重定向到第三方网站，并且带上授权临时票据code参数；
2. 通过code参数加上AppID和AppSecret等，通过API换取access\_token；
3. 通过access\_token进行接口调用，获取用户基本数据资源或帮助用户实现基本操

作。

获取access\_token时序图：



### 3.3 获取code

(1) 修改login.vue，在登陆表单下方添加一个div, 用于显示微信登陆二维码

```
<div id="weixin"></div>
```

(2) 修改login.vue ,引入微信登陆二维码js



```
mounted(){
    var obj = new WxLogin({
        id: "weixin",
        appid: "wx3bdb1192c22883f3",
        scope: "snsapi_login",
        redirect_uri: "http://note.java.itcast.cn/weixinlogin"
    });
},
head:{
    script:[

{src:'http://res.wx.qq.com/connect/zh_CN/htmledition/js/wxLogin.js'}
    ]
}
```

appid: 应用唯一标识

scope: 应用授权作用于

redirect\_uri: 回调地址，是微信登陆成功后要跳转到的页面

(3) 测试: <http://localhost:3000/login> 浏览器显示

微信登录



请使用微信扫描二维码登录  
“传智播客”

我们打开手机用微信扫二维码，会出现以下提示



即将登录传智播客，请确认是本人操作

- 使用你的帐号登录该应用

确认登录

取消

点击确认登陆按钮，浏览器会自动跳转到

[http://note.java.itcast.cn/weixinlogin?  
code=02147Yff12YhgZ0ArCef1qabgf147Yf0&state=undefined](http://note.java.itcast.cn/weixinlogin?code=02147Yff12YhgZ0ArCef1qabgf147Yf0&state=undefined)

这个code是微信发给用户的临时令牌。我们可以根据code再次请求微信第三方登陆接口得到access\_token（正式令牌）

## 3.4 获取access\_token

### 3.4.1 API

通过code获取access\_token

接口说明

通过code获取access\_token的接口。

## 请求说明

```
http请求方式: GET
https://api.weixin.qq.com/sns/oauth2/access_token?
appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code
```

## 参数说明

参数	是否必须	说明
appid	是	应用唯一标识，在微信开放平台提交应用审核通过后获得
secret	是	应用密钥AppSecret，在微信开放平台提交应用审核通过后获得
code	是	填写第一步获取的code参数
grant_type	是	填authorization_code

## 返回说明

正确的返回:

```
{
  "access_token": "ACCESS_TOKEN",
  "expires_in": 7200,
  "refresh_token": "REFRESH_TOKEN", "openid": "OPENID",
  "scope": "SCOPE"
}
```

参数	说明
access_token	接口调用凭证
expires_in	access_token接口调用凭证超时时间，单位（秒）
refresh_token	用户刷新access_token
openid	授权用户唯一标识
scope	用户授权的作用域，使用逗号（,）分隔

## 3.4.2 编写node服务

创建新的node工程 weixinlogin 用于调用微信第三方登陆接口 工程下创建server.js

```
var http = require('http');
var https = require('https');
var url = require('url');
http.createServer(function(request,response){
    var params=url.parse(request.url, true).query;
    var appid='wx3bdb1192c22883f3';
    var secret='db9d6b88821df403e5ff11742e799105';
    if(params.operation==='token'){
        https.get(`https://api.weixin.qq.com/sns/oauth2/access_token?
appid=${appid}&secret=${secret}&code=${params.code}&grant_type=authorizat
ion_code`, function (res) {
            res.on('data', function (chunk) {
                response.writeHead(200,{ 'Content-
Type': 'application/json;charset=utf-8' , "Access-Control-Allow-Origin":
"*" });
                response.end(chunk);
            });
        })
    }
}).listen(8888);
// 终端打印如下信息
console.log('Server running at http://127.0.0.1:8888/');
```

在控制台输入 `node server` 运行服务

地址栏测试: <http://localhost:8888/?code=02147Yff12YhgZ0ArCef1qabgf147Yf0&operation=token>

结果如下:



```
{
  "access_token": "10_zSHADX2JGMivKFfa4nMbZV3ECzY21UY3qrF5ADyjpr_iiLUifo-
n1N0GaRnUEN9T7BagiwSC07awplRFIO1Ghw",
  "expires_in": 7200,
  "refresh_token":
"10__z18gcJz0RXVDKtksbNTQJZ2uK1HiLJZ3I5PcSKA2VB3b6WXi2CR3R_htW6B8kK0mj-
91p08SJMfVKkL84vP1w",
  "openid": "oypcC1u9r-mxVsRGSLFqE65lysVI",
  "scope": "snsapi_login",
  "unionid": "o6JuL1gaIwnVsZC5BpRYImTHKTm8"
}
```

### 3.4.3 调用node服务

node服务编写完成后，我们在十次方前台工程中调用node服务

(1) 编写API,创建api/weixin.js

```
import axios from 'axios'
export default {
  getAccessToken(code){
    return axios.get(`http://localhost:8888?
operation=token&code=${code}`)
  }
}
```

(2) 创建utils/param.js （用于获取浏览器地址栏参数）

```
export function getUrlParam(name) {
  var reg = new RegExp("(^|&)" + name + "=(^[^&]*)(&|$)");
  var r = window.location.search.substr(1).match(reg);
  if(r != null) return unescape(r[2]);
  return null;
}
```

(3) 创建pages/weixinlogin.vue



```
<template>
  <div></div>
</template>
<script>
import {getUrlParam} from '@/utils/param'
import weixin from '@/api/weixin'
import {setUser} from '@/utils/auth'
export default {
  mounted(){
    let code=getUrlParam('code')
    if(code!==null){//如果是微信登陆
      //根据code获取access_token
      weixin.getAccessToken(code).then( res=>{
        let access_token= res.data.access_token
        let openid= res.data.openid
        console.log('access_token:'+access_token+ 'openid:'+openid)
      })
    }
  }
}
```

## 3.5 获取用户昵称与头像

### 3.5.1 API

http请求方式: GET

<https://api.weixin.qq.com/sns/userinfo?>

[access\\_token=ACCESS\\_TOKEN&openid=OPENID](#)

参数	是否必须	说明
access_token	是	调用凭证
openid	是	普通用户的标识，对当前开发者帐号唯一
lang	否	国家地区语言版本，zh_CN 简体，zh_TW 繁体，en 英语，默认为zh-CN



## 返回说明

正确的Json返回结果：

```
{
  "openid": "OPENID",
  "nickname": "NICKNAME",
  "sex": 1,
  "province": "PROVINCE",
  "city": "CITY",
  "country": "COUNTRY",
  "headimgurl":
    "http://wx.qlogo.cn/mmopen/g3MonUZtNHkdmzicIlibx6iaFqAc56vxLSUfpb6n5WKSYVY
    0ChQKkiaJSgQ1dZuTOgvLLRhJbERQQ4eMsv84eavHiaiceqxibJxCfHe/0",
  "privilege": [
    "PRIVILEGE1",
    "PRIVILEGE2"
  ],
  "unionid": " o6_bmasdasdsad6_2sgVt7hMZOPfL"
}
```

参数	说明
openid	普通用户的标识，对当前开发者帐号唯一
nickname	普通用户昵称
sex	普通用户性别，1为男性，2为女性
province	普通用户个人资料填写的省份
city	普通用户个人资料填写的城市
country	国家，如中国为CN
headimgurl	用户头像，最后一个数值代表正方形头像大小（有0、46、64、96、132数值可选，0代表640*640正方形头像），用户没有头像时该项为空
privilege	用户特权信息，json数组，如微信沃卡用户为（chinaunicom）
unionid	用户统一标识。针对一个微信开放平台帐号下的应用，同一用户的unionid是唯一的。

## 3.5.2 编写node服务

修改node工程 weixinlogin的server.js, 新增代码

```
.....
    if(params.operation==='userinfo'){
        https.get(`https://api.weixin.qq.com/sns/userinfo?
access_token=${params.access_token}&openid=${params.openid}`, function
(res) {
            res.on('data', function (chunk) {
                // 发送响应数据 "Hello World"
                response.writeHead(200,{ 'Content-
Type': 'application/json;charset=utf-8' , "Access-Control-Allow-Origin":
"*" });
                response.end(chunk);
            });
        })
    }
}
.....
```

### 3.5.3 调用node服务

(1) 编写API,修改api/weixin.js 新增方法 用于根据access\_token和openid获取用户信息

```
getUserinfo(access_token,openid){  
    return axios.get(`http://localhost:8888?  
operation=userinfo&access_token=${access_token}&openid=${openid}`)  
}
```

(2) 修改pages/weixinlogin.vue

```
import {setUser} from '@/utils/auth'
```

在获取access\_token和openid后，再次请求接口，获取昵称和头像，保存到cookie中

```
mounted(){  
    let code=getUrlParam('code')  
    if(code!==null){//如果是微信登陆  
        //根据code获取access_token  
        weixin.getAccessToken(code).then( res=>{  
            let access_token= res.data.access_token  
            let openid= res.data.openid  
            weixin.getUserinfo( access_token, openid ).then( res => {  
                //提取用户昵称和头像 *****  
                let nickname= res.data.nickname  
                let headimgurl= res.data.headimgurl  
                setUser(access_token,nickname,headimgurl)  
                location.href='/'  
            })  
        })  
    }  
}
```

## 3.6 域名与端口设置

我们刚才都是要通过手动更改url才能完成测试，主要是因为回调地址是域名而我们的工程是本地地址。其实我们要想实现一气呵成的效果也不难，只要通过域名和端口设置即可。

### 3.6.1 域名指向

我们可以通过SwitchHosts 配置域名指向

```
127.0.0.1 note.java.itcast.cn
```

这样我们的工程就可以通过 <http://note.java.itcast.cn:3000>来访问了

### 3.6.2 NUXT端口设置

修改package.json ,添加配置

```
"config": {  
  "nuxt": {  
    "port": "80"  
  }  
},
```

重新启动工程，就可以通过<http://note.java.itcast.cn> 来访问了。

通过以上修改后，我们再次测试微信扫码登陆，就可以看到和生产环境一样的运行效果。

## 4 用户中心嵌套布局

### 4.1 子布局页

(1) 创建pages/manager.vue ,这个为用户中心的布局页



```
<template>
<div>
  <div class="myhome-personinfo" style="background-image:
url('~assets/img/widget-homebg.png');">
    <div class="wrapper">
      <div class="person-baseinfo">
        <!--头像信息-->
        <div class="photo">
          
          <div class="share">
            <span></span>
            <span></span>
            <span></span>
          </div>
        </div>
        <!--文字信息-->
        <div class="info">
          <h1>Web爱好者<span class="allinfo"><a href="~/assets/person-
myfile.html" target="_blank">查看完整档案</a></span></h1>
          <ul class="fill">
            <li> <i class="fa fa-map-marker" aria-hidden="true"></i> <span
class="edit-item"> 填写现居城市</span>
            <form action="" class="sui-form form-inline">
              <input type="text" placeholder="现居城市" />
              <button class="sui-btn btn-danger save-btn">保存</button>
            </form> </li>
            <li> <i class="fa fa-graduation-cap" aria-hidden="true"></i> <span
class="edit-item"> 填写毕业院校</span>
            <form action="" class="sui-form form-inline">
              <input type="text" placeholder="院校名称" />
              <input type="text" placeholder="所学专业" />
              <button class="sui-btn btn-danger save-btn">保存</button>
            </form> </li>
            <li> <i class="fa fa-shopping-bag" aria-hidden="true"></i> <span
class="edit-item"> 填写所在公司/组织</span>
            <form action="" class="sui-form form-inline">
              <input type="text" placeholder="公司/组织名称" />
```



```
<input type="text" placeholder="职位头衔" />
<button class="sui-btn btn-danger save-btn">保存</button>
</form> </li>
<li> <i class="fa fa-link" aria-hidden="true"></i> <span
class="edit-item"> 填写个人网站</span>
  <form action="" class="sui-form form-inline">
    <input type="text" placeholder="个人网站" />
    <button class="sui-btn btn-danger save-btn">保存</button>
  </form> </li>
</ul>
</div>
</div>
<!--右侧编辑-->
<div class="edit-info">
  <h4>个人简介<span class="addedit">编辑</span></h4>
  <div class="info-box">
    <div class="edit-intro">
      暂时没有个人简介
    </div>
  </div>
</div>
<div class="clearfix"></div>
</div>
<!--两列布局-->
<div class="wrapper myhome">
  <div class="left-list">
    <div class="myhome-list">
      <ul class="home-list">
        <li class="active"><a href="~/assets/person-homepage.html">我的主
页</a></li>
        <li><a href="~/assets/person-myanswer.html">我的回答</a></li>
        <li><a href="~/assets/person-myquestion.html">我的提问</a></li>
        <li><a href="~/assets/person-myshare.html">我的分享</a></li>
      </ul>
      <ul class="home-list bottom">
        <li><a href="~/assets/person-dynamic.html">个人动态</a></li>
        <li><a href="~/assets/person-myfocus.html">我的关注</a></li>
        <li><a href="~/assets/person-mycollect.html">我的收藏</a></li>
        <li><a href="~/assets/person-myreaded.html">浏览记录</a></li>
      </ul>
    </div>
  </div>
</div>
```



```
<li><a href="/assets/person-account.html">账户设置</a></li>
</ul>
</div>
</div>
<div class="right-content">
  <nuxt-child/>
</div>
<div class="clearfix"></div>
</div>
</div>
</template>
<script>
import '~/assets/css/page-sj-person-homepage.css'
</script>
```

注意：我们使用 `<nuxt-child/>` 标签

（2）在pages下创建manager文件夹，manager文件夹下创建index.vue（用户中心的默认首页）





```
<template>
  <div class="home-content">
    <ul class="sui-nav nav-tabs nav-large">
      <li class="active"><a href="#one" data-toggle="tab">我的提问</a>
    </li>
    <li><a href="#two" data-toggle="tab">我的回答</a></li>
    </ul>
    <div class="tab-content tab-wrapped">
      <div id="one" class="tab-pane active">
        <ul class="question-list">
          <li> <span class="fl good"><span class="num">12</span> 有用</span>
          <span class="title"><a href="#">有关PHP初级进阶的问题</a></span> <span
          class="fr date">4月6日</span> <span class="clearfix"></span> </li>
          <li> <span class="fl good"><span class="num">12</span> 有用</span>
          <span class="title"><a href="#">有关JAVA初级进阶的问题</a></span> <span
          class="fr date">4月6日</span> <span class="clearfix"></span> </li>
          <li> <span class="fl good"><span class="num">12</span> 有用</span>
          <span class="title"><a href="#">有关HTML5初级进阶的问题</a></span> <span
          class="fr date">4月6日</span> <span class="clearfix"></span> </li>
          <li> <span class="fl good"><span class="num">12</span> 有用</span>
          <span class="title"><a href="#">有关C++初级进阶的问题</a></span> <span
          class="fr date">4月6日</span> <span class="clearfix"></span> </li>
          <li> <span class="fl good"><span class="num">12</span> 有用</span>
          <span class="title"><a href="#">有关python初级进阶的问题</a></span> <span
          class="fr date">4月6日</span> <span class="clearfix"></span> </li>
        </ul>
      </div>
      <div id="two" class="tab-pane">
        <ul class="question-list">
          <li> <span class="fl good"> <span class="num">8</span> 有用</span>
          <span class="title"><a href="#">有关PHP初级进阶的问题</a></span> <span
          class="fr date">2017-07-05 15:08</span> <span class="clearfix"></span>
          </li>
          <li> <span class="fl good"> <span class="num">7</span> 有用</span>
          <span class="title"><a href="#">有关JAVA初级进阶的问题</a></span> <span
          class="fr date">2017-07-05 15:08</span> <span class="clearfix"></span>
          </li>
          <li> <span class="fl good"> <span class="num">6</span> 有用</span>
```



```
<span class="title"><a href="#">有关HTML5初级进阶的问题</a></span> <span
class="fr date">2017-07-05 15:08</span> <span class="clearfix"></span>
</li>
    <li> <span class="fl good"> <span class="num">12</span> 有
用</span> <span class="title"><a href="#">有关C++初级进阶的问题</a></span>
<span class="fr date">2017-07-05 15:08</span> <span class="clearfix">
</span> </li>
    <li> <span class="fl good"> <span class="num">12</span> 有
用</span> <span class="title"><a href="#">有关python初级进阶的问题</a>
</span> <span class="fr date">2017-07-05 15:08</span> <span
class="clearfix"></span> </li>
</ul>
</div>
</div>
<div class="activities">
<h4 class="tit"><span>我的动态</span></h4>
<ul class="activities-content">
<li>
<div class="index-title">
<span class="author">本杰明</span>
<span class="operate">关注了标签</span> &middot;
<span class="time">3小时前</span>
</div>
<div class="guanzhuname">
<span class="tag">php</span>
<span class="tagnum">100</span> 关注
</div>
<div class="intro">
    PHP，是英文超文本预处理语言 Hypertext Preprocessor 的缩写。PHP 是一
种开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入HTML中使用。PHP 的语法借鉴
吸收C语言、Java和Perl等流行计算机语言的特点，易于一般程序员学习。
</div> </li>
<li>
<div class="index-title">
<span class="author">本杰明</span>
<span class="operate">回答了问题</span> &middot;
<span class="time">3小时前</span>
</div>
<div class="question">
<p class="title">网页链接如何直接打开微信，并进入公众号关注页面</p>
<p class="content">现在针对这个微信是屏蔽的，你可以选择通过连接到一个
```



其他的公众号文章中进行关注。</p>

```
</div>
<div class="qa-num">
  <span>关注<i>1</i></span>
  <span>回答<i>2</i></span>
</div> </li>
<li>
  <div class="index-title">
    <span class="author">本杰明</span>
    <span class="operate">收藏了文章</span> &middot;
    <span class="time">3小时前</span>
  </div>
  <div class="question">
    <p class="title">网页链接如何直接打开微信，并进入公众号关注页面</p>
  </div>
  <div class="qa-num">
    <span><a href="#">http://baidu.com</a></span>
  </div> </li>
<li>
  <div class="index-title">
    <span class="author">本杰明</span>
    <span class="operate">收藏了文章</span> &middot;
    <span class="time">3小时前</span>
  </div>
  <div class="question">
    <p class="title">网页链接如何直接打开微信，并进入公众号关注页面</p>
  </div>
  <div class="qa-num">
    <span><a href="#">http://baidu.com</a></span>
  </div> </li>
<li>
  <div class="index-title">
    <span class="author">本杰明</span>
    <span class="operate">回答了问题</span> &middot;
    <span class="time">3小时前</span>
  </div>
  <div class="question">
    <p class="title">网页链接如何直接打开微信，并进入公众号关注页面</p>
    <p class="content">现在针对这个微信是屏蔽的，你可以选择通过连接到一个
其他的公众号文章中进行关注。</p>
  </div>
```

```
<div class="qa-num">
  <span>关注<i>1</i></span>
  <span>回答<i>2</i></span>
</div> </li>
</ul>
</div>
</div>
</template>
```

## 4.2 用户中心各子页面

- (1) 创建pages/manager/myanswer.vue（我的问答）
- (2) 创建pages/manager/myquestion.vue（我的提问）
- (3) 创建pages/manager/myshare.vue（我的分享）
- (4) 创建pages/manager/dynamic.vue（个人动态）
- (5) 创建pages/manager/myfocus.vue（我的关注）
- (6) 创建pages/manager/mycollect.vue（我的收藏）
- (7) 创建pages/manager/myreaded.vue（浏览记录）
- (8) 创建pages/manager/account.vue(账户设置)

## 4.3 菜单样式处理

修改pages/manager.vue中的链接地址



```
<div class="myhome-list">
  <ul class="home-list">
    <router-link to="/manager" active-class="active" tag="li" exact >
<a>我的主页</a></router-link>
    <router-link to="/manager/myanswer" active-class="active" tag="li"
exact ><a>我的回答</a></router-link>
    <router-link to="/manager/myquestion" active-class="active"
tag="li" exact ><a>我的提问</a></router-link>
    <router-link to="/manager/myshare" active-class="active" tag="li"
exact ><a>我的分享</a></router-link>
  </ul>
  <ul class="home-list bottom">
    <router-link to="/manager/dynamic" active-class="active" tag="li"
exact ><a>个人动态</a></router-link>
    <router-link to="/manager/myfocus" active-class="active" tag="li"
exact ><a>我的关注</a></router-link>
    <router-link to="/manager/mycollect" active-class="active"
tag="li" exact ><a>我的收藏</a></router-link>
    <router-link to="/manager/myreaded" active-class="active" tag="li"
exact ><a>浏览记录</a></router-link>
    <router-link to="/manager/account" active-class="active" tag="li"
exact ><a>账户设置</a></router-link>
  </ul>
</div>
```

## 4.4 用户中心鉴权

修改pages/manager.vue代码部分

```
import '~/assets/css/page-sj-person-homepage.css'
import {getUser} from '@/utils/auth'
export default {
  created(){
    if(getUser().name===undefined){
      this.$router.push('/login')
    }
  }
}
```

测试：在未登录的情况下在地址栏输入<http://localhost:3000/manager> 会自动跳转到登录页

修改layouts/default.vue的用户名与头像，修改链接到用户中心

```
<li><a href="/manager" class="notice">{{user.name}}</a></li>
...
<li><a href="/manager" class="homego"></a></li>
```