



招聘信息管理软件详细的实现计划

成 员： 陈先锋 陈星宇 张郦楠

报告日期： 2017 年 11 月 26 日

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 编写人 | 时间 | 修改内容 |
| V1.0 | 张郦楠 | 2017.11.25 | 初步拟定 |

**目录**

[一、数据库的连接 2](#_Toc499386894)

[二、系统实现 3](#_Toc499386895)

[1.登陆界面 3](#_Toc499386896)

[2.用户界面 6](#_Toc499386897)

[三、服务器实现 9](#_Toc499386898)

**招聘系统详细的实现计划**

# 一、数据库的连接

微信小程序无法直接连接数据库，所以一般用wx.request调用后台接口，接口里面连接数据库完成业务逻辑并返回json数据。

具体实现代码：

1. const API\_URL = 'http://localhost:4424/api/'
3. function getApi(url,params){
4. return new Promise((res,rej)=>{
5. wx.request({
6. url:API\_URL+'/'+url,
7. data:Object.assign({},params),
8. header:{'Content-Type': 'application/json'},
9. success:res,
10. fail:rej
11. })
12. })
13. }
15. module.exports = {
16. GetByParams(url,page=1,pageSize=20,search = ''){
17. const params = { start: (page - 1) \* pageSize, pageSize: pageSize }
18. return getApi(url, search ? Object.assign(params, { q: search }) : params)
19. .then(res => res.data)
20. },
21. GetById(url,id){
22. return getApi(url, id)
23. .then(res => res.data)
24. }
25. }

# 二、系统实现

## 1.登陆界面

登陆界面可以注册、输入帐号、输入密码、选择登陆角色



具体实现代码：

index.wxml

1. **<view** class="container"**>**
2. **<view** class="usermotto"**>**
3. **<text** class="user-motto"**>**{{motto}}**</text>**
4. 用户名：
5. **<input** type="text" bindinput="userNameInput"**/>**
6. 密码：
7. **<input** type="text" bindinput="userPasswordInput" password="true"**/>**
8. **<button** bindtap="logIn"**>**登录**</button>**
9. **</view>**
10. **</view>**

index.js

1. var app = getApp()
2. Page({
3. data: {
4. motto: '欢迎登录WXapp',
5. userName:'',
6. userPassword:'',
7. id\_token:'',//方便存在本地的locakStorage
8. response:'' //存取返回数据
9. },
10. userNameInput:function(e){
11. this.setData({
12. userName: e.detail.value
13. })
14. },
15. userPasswordInput:function(e){
16. this.setData({
17. userPassword: e.detail.value
18. })
19. console.log(e.detail.value)
20. },
21. logIn:function(){
22. var that = this
23. wx.request({
24. url: 'http://localhost:8000/admin',
25. data: {
26. username: this.data.userName,
27. password: this.data.userPassword,
28. },
29. method: 'GET',
30. success: function (res) {
31. that.setData({
32. id\_token: res.data.id\_token,
33. response:res
34. })
35. try {
36. wx.setStorageSync('id\_token', res.data.id\_token)
37. } catch (e) {
38. }
39. wx.navigateTo({
40. url: '../components/welcome/welcome'
41. })
42. console.log(res.data);
43. },
44. fail: function (res) {
45. console.log(res.data);
46. console.log('is failed')
47. }
48. })
49. }
50. })

## 2.用户界面

每个用户登陆都有不同的界面，但有着大致相同的功能



具体代码实现：

Wxss

/\*\*app.wxss\*\*/

.container {

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: space-between;

padding: 200rpx 0;

box-sizing: border-box;

}

App.js

App({

onLaunch: function () {

// 调用API获取用户地址;

let that = this;

wx.getLocation({

type: 'wgs84',

success: function(res) {

var latitude = res.latitude;

var longitude = res.longitude; //获取用户的地理位置

that.globalData.latitude = res.latitude;

that.globalData.longitude = res.longitude;

}

})

},

getUserInfo:function(cb){

var that = this

if(this.globalData.userInfo){

typeof cb == "function" && cb(this.globalData.userInfo)

}else{

//调用登录接口

wx.login({

success: function (res) {

that.globalData.code = res.code;

wx.getUserInfo({

success: function (res) {

that.globalData.userInfo = res.userInfo //用户信息对象，不包含 openid 等敏感信息

that.globalData.rawData = res.rawData //不包括敏感信息的原始数据字符串，用于计算签名

that.globalData.signature = res.signature //使用 sha1( rawData + sessionkey ) 得到字符串，用于校验用户信息

that.globalData.encryptedData = res.encryptedData //包括敏感数据在内的完整用户信息的加密数据，详细见加密数据解密算法

that.globalData.iv = res.iv //加密算法的初始向量，详细见加密数据解密算法

typeof cb == "function" && cb(that.globalData.userInfo)

}

})

}

})

}

},

globalData:{

userInfo:null,

logs:false,

rawData:null,

signature:null,

encryptedData:null,

iv:null,

latitude:'',

longitude:''

},

data:{

searchId:'111'

}

})

# 三、服务器实现

小程序框架提供了一个名为wx.request()的方法，用于发起HTTP请求，不过值得注意，小程序对并发请求数量限制在了5个。

wx.request({

url: 'https://your-domain/test',

method: 'GET',

data: {

param1: 'foo' ,

param2: 'bar'

},

header: {

'Content-Type': 'application/json'

},

success: function(res) {

console.log(res.data)

}

})