

小白学react之网页获取微信用户信息



天地会珠海分舵 (/u/adb5a44c71d5) [+ 关注](#)

2016.10.16 16:27* 字数 3243 阅读 3467 评论 2 喜欢 14

(/u/adb5a44c71d5)

react-webpack.png

通过上一篇《小白学react之EJS模版实战 (https://link.jianshu.com?t=http://techgogogo.com/2016/10/react_ejs/)》我们学习了如何通过EJS模版生成我们高定制化的index.html文件。

本篇我们将会继续延续我们的alt-tutorial项目的实战计划，去获取微信扫码用户的信息，并将头像显示在我们页面的右上角上。

最终实战效果将如下所示。

首先根据我们的网站url生成二维码，比如我们可以通过浏览器的FeHelper来生成：

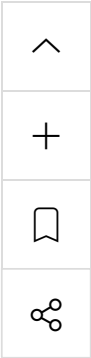


FeHelper.png

然后用户通过微信扫码：

wechat_auth_demo.png

最后用户确定授权后获取到用户的基本信息，并将头像显示在右上角：



avatar_demo.png

1. 内网穿透准备

我们获取微信用户信息的过程中，需要给微信提供回调页面。为了方便调试，我们会将回调页面指向我们本地。这样的话，我们就需要有个办法能让外网(微信服务器)能够直接访问到我们内网中来。

这里我用的内网穿透工具是ngrok，这是海外同行提供的一款内网穿透工具，使用非常方便，大家大可网上查看下该如何配置，这里就不多费唇舌了。

安装好后，通过命令：

```
./ngrok http 8080
```

ngrok就会帮我们将http的8080端口暴露到外网，并为我们分配一个随机的外网访问url地址：

^

+

🔖

🔗

ngrok.jpeg

2. 微信测试号准备

如果您的微信公众号是个人号的话，那么微信将只会给你提供很有限的一些调用接口。所以这里作为个人开发者，我们在调试的时候需要用到微信测试号。

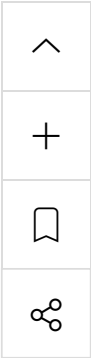
具体怎么进入到微信测试号，我这里也不会多废话，大家自行去搜索下就好了。

这里我们主要有几个事情需要做的：

2.1. 获取微信测试号的appId和appsecret

我们在获取微信用户信息的时候，需要先通过appId获取到code，然后再根据code和appsecret获得access_token，最后根据token获取到用户的信息。

所以我们这里需要获得appId和appsecret。进入微信测试号管理后台后，在左上方我们就可以看到相应的信息：



appId&appsecret.jpeg

2.2. 网页回调域名设置

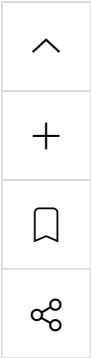
微信用户扫码之后，会弹出授权信息。用户确定授权之后，微信将会根据提供的回调页面进行应用的继续的访问。

我们在授权之后，希望用户能开始访问我们的app，所以我们这里需要设置好回调域名。

在微信测试号后台的同一个管理页面下面，我们可以找到网页服务相关的设置：

网页账号.jpeg

找到“网页授权获取用户基本信息”之后，点击右边的修改会弹出以下页面，我们在此填入我们的回调域名。注意别加上http前缀。这里我们填写的就是上面ngrok给我们生成的公网访问url地址的域名部分：



网页回调域名设置.png

2.3. 关注测试号

通过测试号进行调试的话，我们的测试手机上的微信需要关注该测试号。我们同样可以在测试号管理后台的同一个页面进行扫码关注，达成我们的目的：

关注测试号.jpeg

3. 客户端获取code

3.1. 获取用户信息流程简述

根据微信的《公众平台开发者文档 (<https://link.jianshu.com?t=http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html>)》），通过网页获取微信用户基本信息的流程如下：

- 1. 引导用户进入授权页面同意授权，获取code
- 2. 通过code换取网页授权access_token（与基础支持中的access_token不同）
- 3. 如果需要，开发者可以刷新网页授权access_token，避免过期
- 4. 通过网页授权access_token和openid获取用户基本信息（支持UnionID机制）

作为练习，我们第3步可以跳过。

所以我们这里的第一步是先要去获得code。

根据文档，获取code的时候第一步就是去访问以下的页面：

```
https://open.weixin.qq.com/connect/oauth2/authorize?
appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=SCOPE&state=STATE#wechat_redirect (https://link.jianshu.com?t=https://open.weixin.qq.com/connect/oauth2/authorize?
appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=snsapi_userinfo&state=STATE#wechat_redirect)。
```

其中我们要提供的有appid和回调地址redirect_uri，其他部分保持不变。

3.2. 授权页面url生成

这里我们提供一个方法来生成这一长段url地址：

```
function generateGetCodeUrl(redirectURL) {  
  
  return new URI("https://open.weixin.qq.com/connect/oauth2/authorize")  
    .addQuery("appid", confidential.APP_ID)  
    .addQuery("redirect_uri", redirectURL)  
    .addQuery("response_type", "code")  
    .addQuery("scope", "snsapi_userinfo")  
    .addQuery("response_type", "code")  
    .hash("wechat_redirect")  
    .toString();  
  
};
```

前面说的我们需要提供的两个数据中，回调页面地址我们通过参数redirectURL传入，appid我将其封装到一个独立的文件config/Confidential.js中，大家根据自己的情况填写就好了(需要注意的是，该文件我会在.gitignore文件中配置成不上传到github，所以大家记得自己自行添加)。

```
const confidential = {  
  APP_ID: 'xxxxxx', //Please use your owe app id;  
  APP_SECRET: 'xxxxxxxxx', //Please use your owe secret  
};  
  
export default confidential;
```

3.3. 进入根路由时引导用户进入授权页

封装好微信授权页面后，我们就要考虑应该在什么时间点引导用户进入授权页面了。

我这里是希望用户在扫码之后立刻进入授权页面，所以我们应该在进入第一个页面之前就打开引导页。

那么如何做到呢？这里我们需要用到react router的onEnter属性方法，通过这个方法，当用户访问某个路由之前，会先执行onEnter指定的方法。

比如我们希望在进入根路由后先检查用户授权：



```

class RootRouters extends Component {
  ...
  render() {
    const { history } = this.props;
    return (
      <Router history = {history} >
        <Route name='index' path="/" onEnter={this.wechatAuth.bind(this)} component={IndexRoute} />
        <Route name="about" path="/about" component={About} />
        <Route name="locations" path="/locations" component={Locations} />
      </Router>
    );
  }
  ...
}

```

那么当用户访问<http://localhost:8080> (<https://link.jianshu.com?t=http://localhost:8080>) 或者从外网访问ngrok帮我们生成的外网url(我这里是<http://79214cd7.ngrok.i> (<https://link.jianshu.com?t=http://79214cd7.ngrok.i>))的时候, 就会先去调用wechatAuth这个方法。

```

class RootRouters extends Component {
  ...
  wechatAuth(nextState, replace, next) {

    const uri = new URI(document.location.href);
    const query = uri.query(true);
    const {code} = query;

    if(code) {
      WechatUserStore.fetchUserInfo(code);

      next();
    } else {

      document.location = generateGetCodeUrl(document.location.href);
    }

  }
  ...
}

```

在wechatAuth这个方法中, 最关键的就是后面这一句:

```
document.location = generateGetCodeUrl(document.location.href);
```

其目的就是当前页面的url作为回调页面传入到上面的generateGetCodeUrl里面, 生成授权页面url, 然后将其赋值给页面的document.location, 其结果就是当前页面会被授权页面覆盖掉, 其呈现效果就是用户将会进入到本文最上面提及的授权页面。



当用户授权之后，就会重定向到回调页面，也就是我们的ngrok生成的外网访问url，且这时候该url会通过query的方式带上要传过来的code比如：`http://79214cd7.ngrok.io?code=xxxxxx&state=STATE` (`https://link.jianshu.com?t=http://79214cd7.ngrok.io?code=xxxxxx&state=STATE`)。其中xxxxxx就是code的值。

这样就又会再次访问我们的根由路，也就会再次进入到wechatAuth这个方法里面，因为这次通过微信回调回来的访问中query会带有code的信息，所以这该段代码前面会通过urijs包的功能来先把code解析出来。

这个时候我们就会判断这个code是否存在，如果不存在(用户手动扫码访问页面的时候没有带query，所以这个code不存在)的话，就重定向到微信授权页面；存在的话，就通知服务器去根据code获取用户的基本信息。

3.4. 客户端发送code到服务端请求用户信息

这里跟服务器的沟通我们封装到alt框架的Store里面，整一套alt的Actions，Store和Source的构建，通过我们之前的学习，我相信我们已经是驾轻就熟的了。这里我们主要看下Source中是如何和服务端沟通的就好了，其他有什么不清楚的大家可以通过文章后面的描述直接查看源码，或者往回翻下我网站上关于小白学react的系列文章。



```
import defaults from 'superagent-defaults';
import superagentPromisePlugin from 'superagent-promise-plugin';
var WechatUserActions = require('../actions/WechatUserActions');
import co from 'co';
const request = superagentPromisePlugin(defaults());

var WechatUserSource = {
  fetchUserInfo() {
    return {
      remote(state,code) {
        return co(function *() {
          let userInfo = null;
          const getUserInfoUrl = `/api/user_info?code=${code}`;
          try {
            let result = yield request.get(getUserInfoUrl);
            userInfo = result.text;
          } catch (e) {
            userInfo = null;
          }
          //console.log("userInfo:", userInfo);
          return userInfo;
        });
      },

      local() {
        // Never check locally, always fetch remotely.
        return null;
      },

      success: WechatUserActions.updateUserInfo,
      error: WechatUserActions.getUserInfoFailed,
      loading: WechatUserActions.getUserInfo,
    }
  }
};

module.exports = WechatUserSource;
```

这里跟服务端的沟通其实和之前的LocationSource没有太大区别，同样是发送一个get的request请求 `/api/user_info?code=${code}` 到服务器端，并在request的url中带上code这个query。当服务端通过code获取到用户信息之后，再返回来就出发WechatUserActions的updateUserInfo这个Action。跟着这个Action就会出发WechatUserStore中的onUpdateUserInfo的方法：



```
class WechatUserStore {
  constructor() {
    this.userInfo = [];
    this.errorMessage = null;

    this.bindActions(WechatUserActions);

    this.exportAsync(WechatUserSource);
  }
  onUpdateUserInfo(userInfo) {
    this.userInfo = JSON.parse(userInfo);
    this.errorMessage = null;
  }
  ...
}
```

这个方法会将返回来的用户信息字符串转化成json格式，然后保存起来本WechatUserStore的userInfo成员变量上面。

4. 服务器端根据code获取用户信息

现在客户端已经获取到code，并将code通过 `/api/user_info?code=${code}` 请求传到服务器端。那么服务器端要做的事情就是根据这个code来获得对应的用户信息。

4.1. 服务端根据code和appsecret获取access_token和openid

其中服务器要做的第一步事情就是根据客户端传过来的code获取到访问用户信息的access_token。

那么我们这里首要的就是要在express服务端应用中加入"/api/user_info"的访问路由：

```
app.get("/api/user_info", function(req,res) {
  ...
})
```

然后在该代码里面开始获取access_token的信息。

根据微信开发文档，获取该access_token的请求url的格式如下：

```
https://api.weixin.qq.com/sns/oauth2/access_token?
appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code
(https://link.jianshu.com?t=https://api.weixin.qq.com/sns/oauth2/access_token?
appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code)
```

这里我们需要同时提供appId，appsecret和刚才获得的code。所以我们这里只需要按需求填进去就好了：



```
const code = req.query.code;
const getTokenUrl = `https://api.weixin.qq.com/sns/oauth2/access_token?appid=${confide
```

填好之后我们就可以通过superAgent来将该请求发送到微信服务器并读取返回了：

```
co ( function *() {
  try {
    let result = yield request.get(getTokenUrl);
    tokenInfo = JSON.parse(result.text);
  } catch (e) {
    console.log("exception on getting access token");
    tokenInfo = null;
  }
  console.log("token info:", tokenInfo);
  ...
}
```

从官方提供的开发文档可以看到，最终返回的数据有：

```
{
  "access_token": "ACCESS_TOKEN",
  "expires_in": 7200,
  "refresh_token": "REFRESH_TOKEN",
  "openid": "OPENID",
  "scope": "SCOPE",
  "unionid": "o6_bmasdasdsad6_2sgVt7hMZOPfL"
}
```

其中access_token就是我们下面获取用户需要用到的token，而openid就是该访问用户的唯一id信息，这就是我们的目标用户。

4.2 服务端根据access_token和openid获取用户信息

获取到access_token和用户的openid之后，我们就可以开始获取用户的基本信息了。

获取用户的基本信息的请求uri格式如下：



```
https://api.weixin.qq.com/sns/userinfo?  
access_token=ACCESS_TOKEN&openid=OPENID&lang=zh_CN  
(https://link.jianshu.com?t=https://api.weixin.qq.com/sns/userinfo?  
access_token=ACCESS_TOKEN&openid=OPENID&lang=zh_CN)
```

这个请求uri比起前面获取access_token的稍微简单点。我们只需要传入access_token和用户的openid就ok了。整个请求和解析流程代码如下：

```
if( tokenInfo != null ) {  
  const getUserInfoUrl = `https://api.weixin.qq.com/sns/userinfo?access_token=  
  try {  
    const result2 = yield request.get(getUserInfoUrl);  
    userInfo = JSON.parse(result2.text);  
    console.log("userInfo:",userInfo);  
  } catch(e) {  
    console.log("Exception on getting user info");  
    userInfo = null;  
  }  
  console.log("userInfo:",userInfo);  
}  
  
if (userInfo) {  
  res.send(userInfo);  
}
```

当我们前面获取的access_token没有任何问题的时候，我们就会根据该token和用户的openid发送请求到微信服务器来获取用户基本信息。

当成功获取到用户的基本信息，最后就将其返回给对应的请求客户端。

5. 客户端渲染用户信息

为了能将用户的头像在所有页面都进行显示，我们这里将头像的显示实现在Home页面里面。

我们的Home页面之前只是显示前面的几个标签而已，随着我们慢慢将更多的内容加到Home页面，起维护起来必然会变得困难。

所以这里我们顺便将src/components/Home.jsx的代码稍微重构下。将头部导航标签，显示内容，头像，分开不同的组件，以方便维护：



```

import React from 'react'
import { Link } from 'react-router'
import './Home.scss'
import BaseLayout from "../BaseLayout.jsx";
import WechatUserStore from "../stores/WechatUserStore";
var AltContainer = require('alt-container');

//src='http://wx.qlogo.cn/mmopen/Q3auHgzwzM7H0y8WwY1gCyH8PW2DEhY0S3Rz44cn8TJpGwNWYRuYb

class Headers extends React.Component{
  render() {
    console.log("props.in header:", this.props);

    return (
      <div>
        <nav >
          <li className="home__tab__li"><Link to="/locations">名胜古迹</Link><
          <li className="home__tab__li"><Link to="/about">关于techgogogo</Lin
        </nav>
      </div>
    )
  }
}

class Avatar extends React.Component{
  render() {
    console.log("props.in avatar:", this.props);
    if (WechatUserStore.isLoading()) {
      return (
        <div className="home__avatar">
          
        </div>
      )
    }

    return (
      <div className="home__avatar">
        <img
          src={this.props.userInfo.headimgurl}/>
        </div>
      )
    }
  }

class Contents extends React.Component{
  render() {
    console.log("props.in content:", this.props);

    return (
      <div>
        <div style={{clear: "both"}}></div>
        {this.props.contents}
      </div>
    )
  }
}

class Home extends React.Component{

  render() {
    return (

```



```
<BaseLayout title="Home" style={{"backgroundColor": "white"}}>

  <Headers/>

  <AltContainer store={WechatUserStore}>
    <Avatar />
  </AltContainer>

  <div>
    <Contents contents={this.props.children}> </Contents>
  </div>
</BaseLayout>

)
}
}

module.exports = Home;
```

这样看起来就清晰多了，且修改任何一个组件都不会影响到其他组件，从而让代码更吻合上一篇文章提到的OCP原则。

这里新增加的微信用户头像组件就是Avatar这个Component。起渲染逻辑就是，当用户信息还没有从服务端取到的时候，显示的是loading的gif图片；当已经获得了用户信息之后，显示的就是微信用户的头像。

最后显示的位置和样式我们通过在Home.scss里面增加avatar样式来实现:

```
&__avatar {
  border-radius: 50%;
  border: rem(19px) solid white;
  margin: rem(5px) rem(20px);
  height: rem(260px);
  width: rem(260px);
  background: #d3d3d3;
  overflow: hidden;
  float:right;
  img {
    width: 100%;
    height: 100%;
  }
}
```

设计的基本考虑就是将圆形显示该图片，且浮动在页面的右上角。具体效果请查看文章开始时候的demo图片。

5. 源码



```
git clone https://github.com/kzlathander/alt-tutorial-webpack.git
(https://link.jianshu.com?t=https://github.com/kzlathander/alt-tutorial-webpack.git)
cd alt-tutorial-webpack
checkout 09
npm install
npm run build
```

同时

本文由天地会珠海分舵编写，转载需授权，喜欢点个赞，吐槽请评论，进一步交流请关注本人天地会珠海分舵 (<https://www.jianshu.com/users/adb5a44c71d5/>)以及《微信小程序开发 (<https://www.jianshu.com/collection/e1c18501738e>)》主题。

《未完待续》

小礼物走一走，来简书关注我

赞赏支持

📖 微信公众号TechGoGoGo (/nb/544067)

举报文章 © 著作权归作者所有



天地会珠海分舵 (/u/adb5a44c71d5)

+ 关注

写了 482243 字，被 1218 人关注，获得了 4360 个喜欢
(/u/adb5a44c71d5)

关注产品 · 创业 · 创意 · 科技 · 海外！ 公众号：techgogogo 个人微信：zhubaitian1 www.techgogogo.com

喜欢 | 14



更多分享

^

+

🔖

🔗

被以下专题收入，发现更多相似内容

+ 收入我的专题



微信小程序开发 (/c/e1c18501738e?utm_source=desktop&utm_medium=notes-included-collection)



首页投稿 (暂停... (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)



React (/c/5e54056f1aea?utm_source=desktop&utm_medium=notes-included-collection)



react.js (/c/9df924623a39?utm_source=desktop&utm_medium=notes-included-collection)

【转】微信公众平台开发 OAuth2.0网页授权认证 (/p/52096ed0dc53?utm_...

一、什么是OAuth2.0 官方网站：http://oauth.net/ http://oauth.net/2/ 权威定义：OAuth is An open protocol to allow secure authorization in a simple and sta...



一个被床封印的程序猿 (/u/6186d0924419?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

(/p/5e0644528a34?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendati

微信公众号开发：获取openId和用户信息 (/p/5e0644528a34?utm_campai...

title: 微信公众号开发：获取openId和用户信息 tags: 微信公众号 categories: 笔记 date: 2017-05-01 14:35:34

最近一直在做微信公众号项目的开发，会有一种感觉，就是微信相关的开发真TM的啰嗦麻烦啊。当首次接...



行径行 (/u/f83bd1cdcf0?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

微信开发入门者 学会网页获取用户基本信息 (/p/35b9fd06082c?utm_camp...

如果用户在微信客户端中访问第三方网页，公众号可以通过微信网页授权机制，来获取用户基本信息，进而实现业务逻辑。关于网页授权回调域名的说明 1、在微信公众号请求用户网页授权之前，开发者需要先到...



老牛圣斗士 (/u/d8c24877c11b?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat




(/p/e36a5af3daddb?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendati
微信公众平台开发 OAuth2.0网页授权 (/p/e36a5af3daddb?utm_campaign=...

摘自方倍工作室 一、什么是OAuth2.0官方网站：http://oauth.net/ http://oauth.net/2/权威定义：OAuth is An open protocol to allow secure authorization in a simple...

 jew\ (/u/7a66106d8bce?


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

(/p/34c3714c7375?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendati
微信公众号网页授权summer版 (/p/34c3714c7375?utm_campaign=male...


很早前就想了解下微信公众号开发，懒和拖延症让这计划迟迟没落实，正巧新项目有了微信公众号的业务，我心中没数地回应，我来！顿时，无形的压力向上冒，不知何从下手，果然是矛盾体质，好奇向往新事物...

 英文名叫夏天 (/u/a11824f536ea?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

梦 (/p/fbd530057da9?utm_campaign=maleskine&utm_content=note&...

金色的太阳 炙烤着青春的脸庞 我们迈着大步 踏在炽热的地上 一路小跑 追逐着梦想 快看 梦就在不远的地方 继续奔跑 就能接近梦想

 涛心依旧 (/u/4f114d376a4a?


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

(/p/16a88a85234d?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendati
想起你~ (/p/16a88a85234d?utm_campaign=maleskine&utm_content=...

黑板上还有未做完的选择题 答案却被你悄悄地写在手心 你是否会记起 我抢你的橡皮 到现在竟没有机会还你 不得了，已经是00后的天下了 从稚嫩的小孩，已经慢慢变成了火遍全国的团体 他们的努力，得到了肯定，...

 紫看片子 (/u/587ddf263ed3?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

初遇 (/p/6494996b53b2?utm_campaign=maleskine&utm_content=not...

当钢琴声从一个豪华房间中传出，那是一个穿着精致白色礼服的少年，这一刻夏瑞呆住了，琴声停了，少年转过身，夏瑞一个致命般的微笑，牵动了少年的心，这是沈星阳第一次主动要求留下一个人。夏瑞说是沈...



小孟孟Xmm0621 (/u/ce99ce622ed9?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

(/p/c7dd40bf40a4?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendati

益好康舒复贴：巧克力真的能治女性痛经吗？ (/p/c7dd40bf40a4?utm_ca...

痛经是指女性在月经来潮时的疼痛，一般分为原发性痛经和继发性痛经两种。原发性痛经一般没有明显的盆腔病变，经期的疼痛是由于子宫收缩与或缺血所致，一般能正常排卵，这种痛经通常是从青春开始，但...



益好康舒复贴 (/u/d0a20f8d7ee8?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendat

^

+

🔖

🔗