微信小程序之登录

直接上图，后面再写详细代码，及解释

经过我深思熟虑，直接上代码，看的懂的自然懂，哈哈，根据流程找代码就可以了

**一．入口文件pages/index/index.js onLoad或者OnShow检查session\_key**

import auth from "../../utils/auth";

Page({

data: {

},

//页面加载回调

onLoad: function(options) {

auth.checkSession();

},

});

**二．下面上auth.js文件**

/\*\*

\* 微信登录验证 需要登录的页面使用

\*/

import http from "./request";

const auth = {

/\*\*

\* 检查微信登录session\_key是否过期

\* 1.在需要获取敏感用户信息数据的时候，检查

\* 2.再重新登陆之前检查session\_key

\*/

checkSession: function() {

wx.checkSession({

success() {

//未过期，同时，检查token 是否存在，不存在也要重新执行登录流程,这里的token是和自己服务器换取信息重要凭证 这里其实还可以检查token的合法性 不合法token则重新登录 也可以在需要用token的时候返回错误代码，执行重新登录

wx.getStorage({

key: "token",

fail() {

// 不存在

auth.checkAuth();

}

});

},

fail() {

// session\_key 已经失效，需要重新执行登录流程

auth.checkAuth("expire");

}

});

},

/\*\*

\* 检查是否授权 未授权则跳转到授权页

\* @param {\*传session是否过期} type

\*/

checkAuth: function(type = "") {

wx.getSetting({

success(res) {

if (res.authSetting["scope.userInfo"]) {

//已授权,则开始登录

auth.doLogin({ type: type, back: false });

} else {

//未授权

wx.navigateTo({

url: "/pages/auth\_login/auth\_login?type=" + type

});

}

}

});

},

/\*\*

\* 执行登录操作

\* @param {Object} param0

\*/

doLogin: function({ type, back = true }) {

wx.login({

success(res) {

if (res.code) {

wx.getUserInfo({

lang: "zh\_CN",

success(userInfo) {

// 发起网络请求

http.post({

url: "后端处理url",

data: {

code: res.code,

signature: userInfo.signature,

encryptedData: userInfo.encryptedData,

iv: userInfo.iv,

type: type

},

showLoading: false,

success: function(res) {

if (res.data.code === 0) {

//设置token

wx.setStorage({

key: "token",

data: res.data.data

});

if (back) {

wx.navigateBack({

delta: 1

});

} else {

//刷新当前页

const pages = getCurrentPages();

const perpage = pages[pages.length - 1];

perpage.onShow();

}

} else {

wx.showToast({

title: "登录失败！",

icon: "none"

});

}

}

});

}

});

} else {

wx.showToast({

title: "登录失败！",

icon: "none"

});

}

}

});

}

};

export default auth;

**三．下面是request代码**

import auth from "./auth";

let app = getApp();

let baseURL =

app.globalData.production === "1"

? "正式接口url前缀"

: "测试接口url前缀";

const request = {

get: function({

url,

params = {},

success,

showLoading = true,

isAuth = false

}) {

if (showLoading) wx.showLoading({ title: "加载中" });

request.createHeader(isAuth, function(header) {

wx.request({

url: baseURL + url,

method: "get",

dataType: "json",

data: params,

header: header,

success: function(res) {

if (

typeof res.header.Code !== "undefined" &&

res.header.Code === "40001"

) {

//token错误

wx.removeStorage({

key: "token",

success(res) {

auth.checkSession();

}

});

}

if (typeof res.header.Token !== "undefined") {

wx.setStorage({

key: "token",

data: res.header.Token

});

}

success(res);

},

complete: function() {

wx.hideLoading();

}

});

});

},

post: function({

url,

data = {},

success,

showLoading = true,

isAuth = false

}) {

if (showLoading) wx.showLoading({ title: "加载中" });

request.createHeader(isAuth, function(header) {

wx.request({

url: baseURL + url,

method: "post",

dataType: "json",

data: data,

header: header,

success: function(res) {

if (

typeof res.header.Code !== "undefined" &&

res.header.Code === "40001"

) {

//token错误

wx.removeStorage({

key: "token",

success(res) {

auth.checkSession();

}

});

}

if (typeof res.header.Token !== "undefined") {

wx.setStorage({

key: "token",

data: res.header.Token

});

}

success(res);

},

complete: function() {

wx.hideLoading();

}

});

});

},

createHeader: function(isAuth, compete) {

let header = {

"content-type": "application/json;charset=UTF-8" // 默认值

};

if (isAuth) {

wx.getStorage({

key: "token",

success(res) {

//token存在 但是session\_key过期 这里不影响因为用不到session\_key

header.Auth = res.data;

compete(header);

},

fail() {

// 不存在

auth.checkSession();

}

});

} else {

compete(header);

}

}

};

export default request;

**4.下面是授权页面的js文件 auto\_login.js**

import auth from "../../utils/auth";

Page({

data: {

canIUse: wx.canIUse("button.open-type.getUserInfo"),

type: ""

},

onLoad: function(options) {

this.setData({

type: options.type

});

},

bindGetUserInfo: function(e) {

if (e.detail.errMsg !== "getUserInfo:ok") {

//取消授权等原因

wx.navigateBack({

delta: 1

});

}

let \_this = this;

auth.doLogin({

type: \_this.data.type

});

}

});

**5.最为期待的后端代码，啦啦啦结束**

<?php  
namespace app\index\controller;  
use app\index\common\Base;  
use think\Controller;  
use think\Db;  
use think\facade\Request;  
use think\Exception;  
  
class WeChat extends Base {  
  
 private $appId = "你的appid";  
 private $secret = "你的秘钥去你的小程序后台获取";  
  
 public function doLogin(){  
 $code = 0;  
 $msg = 'success';  
 $data = null;  
 try{  
 if(Request::isPost()){  
 $param = Request::param();  
 if(!isset($param['code'])||!isset($param['signature'])||  
 !isset($param['encryptedData'])||!isset($param['iv'])||!isset($param['type'])){  
 $code = 10002;  
 throw new Exception('缺少参数');  
 }  
  
 //从微信接口获取openid session\_key  
 $url = "https://api.weixin.qq.com/sns/jscode2session?"  
 ."appid={$this->appId}&secret={$this->secret}&js\_code={$param['code']}&grant\_type=authorization\_code";  
 $response = json\_decode(http\_get($url));  
 if(isset($response->errcode)&&$response->errcode!=0){  
 $code = 10003;  
 throw new Exception('登录失败');  
 }else{  
 //获取用户信息，此处用到解密方法  
 $pc=new \weChatLogin\WXBizDataCrypt($this->appId,$response->session\_key);  
 $errCode = $pc->decryptData($param['encryptedData'], $param['iv'], $userInfo );  
 if($errCode!==0){  
 $code = 10004;  
 throw new Exception('解密用户数据失败');  
 }  
 //记录openID sessionKey 生成加密token返回  
 $user = Db::name('微信用户表')  
 ->where([  
 'openId'=>$response->openid  
 ])  
 ->find();  
 if($user){  
 if($param['type'] === 'expire'){  
 //session\_key过期了 需要更新session\_key  
 $update = [  
 'sessionKey'=>$response->session\_key,  
 ];  
 Db::name('微信用户表')  
 ->where([  
 'openId'=>$response->openid,  
 ])  
 ->update($update);  
 }  
 $token= $user['token'];  
 }else{  
 $userInfo = json\_decode($userInfo);  
 $insert = [  
 'nickName'=>base64\_encode($userInfo->nickName),  
 'city'=>$userInfo->city,  
 'province'=>$userInfo->province,  
 'country'=>$userInfo->country,  
 'avatarUrl'=>$userInfo->avatarUrl,  
 'openId'=>$response->openid,  
 'sessionKey'=>$response->session\_key,  
 ];  
 $userId = Db::name('微信用户表')  
 ->insertGetId($insert);  
 $token = $this->refreshUserToken($userId);  
 }  
 $data = $token;  
 }  
  
 }else{  
 $code = 10001;  
 throw new Exception('请求失败');  
 }  
  
 }catch (Exception $e){  
 $msg = $e->getMessage();  
 }  
 return json(['data'=>$data,'msg'=>$msg,'code'=>$code]);  
 }  
  
 public static function refreshUserToken($userId){  
 if(!$userId)  
 return false;  
 //生成token  
 $rand\_num = rand(10,99999);//随机数  
 $time = time();//时间戳  
 $token = md5($userId.$rand\_num.$time);  
 $expire\_time = time() + 600;//过期时间为10分钟  
 //更新当前用户token 和 有效期  
 $update = [  
 'token'=>$token,  
 'expire\_time'=>$expire\_time,  
 ];  
 $update\_res = Db::name('微信用户表')->where(['id'=>$userId])->update($update);  
 if(!$update\_res)  
 return false;  
 else  
 return $token;  
 }  
}

另外关于token怎么维护，这里就不放出来了，若有疑问，请发问题至邮箱。