/\*

微信支付方法(uni-app h5)适用

获取微信加签信息

@param{data}:获取的微信加签

@param{res}:成功回调

@param{fail}:失败回调

@warn:因为package为严格模式下的保留字，不能用作变量.

@use

wPay({

appId,

timeStamp,

nonceStr,

signature,

package,

paySign

},res=>{

console.log(‘调用成功!‘);

},fail=>{

console.log(‘调用失败!‘);

})

\*/

const wx = require('jweixin-module');

class WexinPayFactory {

wexinPay(data, successCallBack) {

let [appId, timestamp, nonceStr, signature, packages, paySign] = [data.appId, data.timeStamp, data.nonceStr, data.sign,

data.package, data.sign

];

wx.config({

debug: false, // 开启调试模式,调用的所有api的返回值会在客户端alert出来，若要查看传入的参数，可以在pc端打开，参数信息会通过log打出，仅在pc端时才会打印。

appId, // 必填，公众号的唯一标识

timestamp, // 必填，生成签名的时间戳

nonceStr, // 必填，生成签名的随机串

signature, // 必填，签名，见附录1

jsApiList: ['chooseWXPay'] // 必填，需要使用的JS接口列表，所有JS接口列表见附录2

});

wx.ready(function() {

wx.chooseWXPay({

timestamp, // 支付签名时间戳，注意微信jssdk中的所有使用timestamp字段均为小写。但最新版的支付后台生成签名使用的timeStamp字段名需大写其中的S字符

nonceStr, // 支付签名随机串，不长于 32 位

'package': packages, // 统一支付接口返回的prepay\_id参数值，提交格式如：prepay\_id=\*\*\*）

signType: 'MD5', // 签名方式，默认为‘SHA1‘，使用新版支付需传入‘MD5‘

paySign, // 支付签名

success(res) {

// 支付成功后的回调函数

successCallBack(res);

},

fail(res) {

uni.showToast({

icon: 'none',

title: '支付失败'

})

return;

}

});

});

wx.error(function(res) {

// config信息验证失败会执行error函数，如签名过期导致验证失败，具体错误信息可以打开config的debug模式查看，也可以在返回的res参数中查看，对于SPA可以在这里更新签名。

/\*alert("config信息验证失败");\*/

});

}

};

module.exports = new WexinPayFactory();