1. 配置fiexablejs在index.html中添加以下js

<script>

!(function(win, doc) {

function setFontSize() {

var baseFontSize = 100;

var baseWidth = 320;

var clientWidth = document.documentElement.clientWidth || window.innerWidth;

var innerWidth = Math.max(Math.min(clientWidth, 480), 320);

var rem = 100;

if (innerWidth > 362 && innerWidth <= 375) {

rem = Math.floor(innerWidth / baseWidth \* baseFontSize \* 0.9);

}

if (innerWidth > 375) {

rem = Math.floor(innerWidth / baseWidth \* baseFontSize \* 0.84);

}

window.\_\_baseREM = rem;

document.querySelector('html').style.fontSize = rem + 'px';

}

var evt = 'onorientationchange' in win ? 'orientationchange' : 'resize';

var timer = null;

win.addEventListener(evt, function() {

clearTimeout(timer);

timer = setTimeout(setFontSize, 300);

}, false);

win.addEventListener("pageshow", function(e) {

if (e.persisted) {

clearTimeout(timer);

timer = setTimeout(setFontSize, 300);

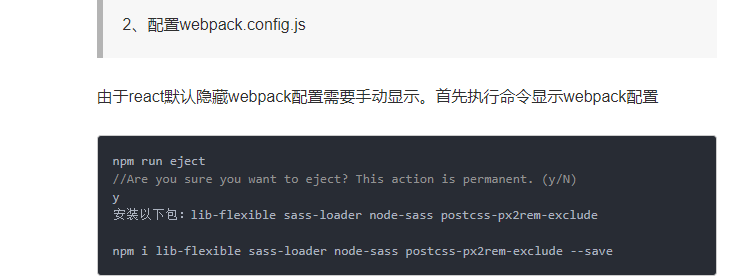
}

}, false);

setFontSize();

}(window, document));

</script>

1. 

pm run eject

//Are you sure you want to eject? This action is permanent. (y/N)

y

安装以下包：lib-flexible sass-loader node-sass postcss-px2rem-exclude

npm i lib-flexible sass-loader node-sass postcss-px2rem-exclude --save

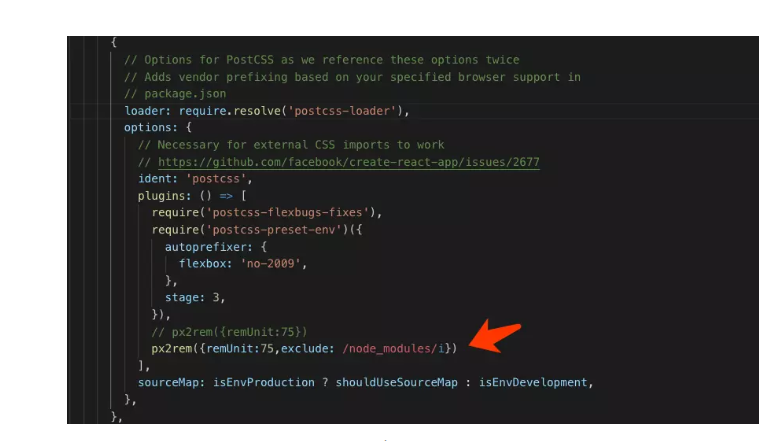
修改webpack.config.js ，首先引入：

const px2rem = require('postcss-px2rem-exclude');

//加入

px2rem({remUnit:75,exclude: /node\_modules/i})

然后找到处理postcss-loader的地方：



在我们的入口文件index.js 中引入lib-flexible

import "lib-flexible"

最后一步修改index.html

<meta content="width=device-width,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no" name="viewport" />

参考地址：

<https://www.jianshu.com/p/285d51730f5e>

1. Vscode-eslint 格式化：

Mac 下setting =》open-setting=》

{

  // #每次保存的时候自动格式化

  "editor.formatOnSave": true,

  // #每次保存的时候将代码按eslint格式进行修复

  "eslint.autoFixOnSave": true,

  // 添加 react,vue 支持

  "eslint.validate": [

    "javascript",

    "javascriptreact",

    {

      "language": "vue",

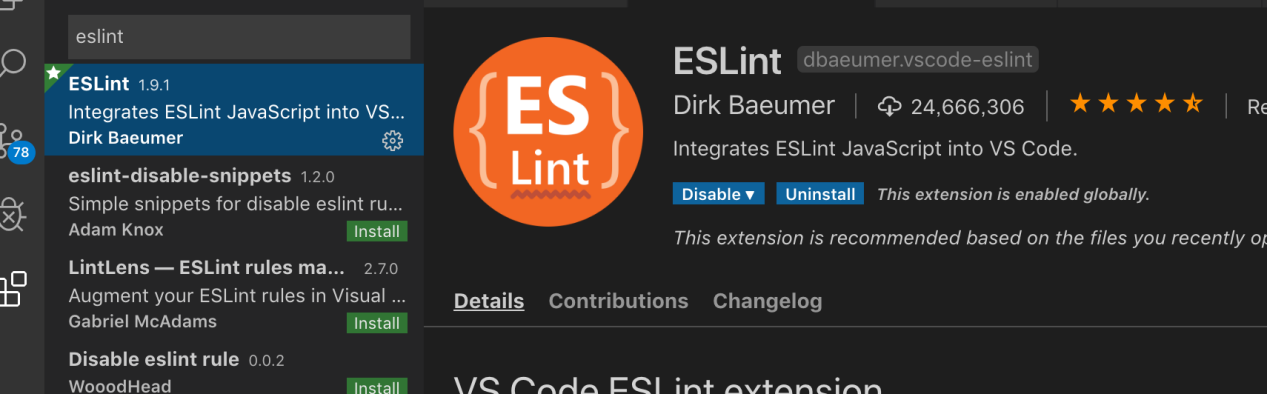
      "autoFix": true

    }

  ]

}

1. 1在vscode中安装eslint 插件

3.2 

3.3 项目根目录配置.eslintrc.js

/\*

 \*    Copyright (c) 2018-2025, binsoo All rights reserved.

 \*

 \* Redistribution and use in source and binary forms, with or without

 \* modification, are permitted provided that the following conditions are met:

 \*

 \* Redistributions of source code must retain the above copyright notice,

 \* this list of conditions and the following disclaimer.

 \* Redistributions in binary form must reproduce the above copyright

 \* notice, this list of conditions and the following disclaimer in the

 \* documentation and/or other materials provided with the distribution.

 \* Neither the name of the www.gac-nio.com developer nor the names of its

 \* contributors may be used to endorse or promote products derived from

 \* this software without specific prior written permission.

 \* Author: binsoo (chenfengbin@gac-nio.com)

 \*/

module.exports = {

  root: true,

  parser: "babel-eslint",

  parserOptions: {

    sourceType: "module"

  },

  env: {

    browser: true,

    node: true,

    es6: true

  },

  extends: "eslint:recommended",

  // required to lint \*.vue files

  plugins: ["html"],

  // check if imports actually resolve

  settings: {

    "import/resolver": {

      webpack: {

        config: "build/webpack.base.conf.js"

      }

    }

  },

  // add your custom rules here

  //it is base on https://github.com/vuejs/eslint-config-vue

  rules: {

"no-console": "off",

"no-debugger": false

  }

};

3.4 vue-cli 项目在pagejson.js 下面配置

 "eslintConfig": {

    "root": true,

    "env": {

      "node": true

    },

    "extends": [

      "plugin:vue/essential",

      "eslint:recommended"

    ],

    "rules": {

      "no-console": "off",

"no-debugger": false

    },

    "parserOptions": {

      "parser": "babel-eslint"

    }

  },

# 4.[小程序更新发布之后线上还是之前的老版本](https://www.cnblogs.com/520BigBear/p/11262164.html)

微信小程序每次发布更新后线上还是之前的老版本，**要删除后重新打开才会显示新版本**，特烦。

不知道是不是太久没去看文档了，发现 api 中增加了更新 wx.getUpdateManager()，于是立马来试试。

在 app.js 中增加一个 onLaunch 生命周期函数，然后把微信的这段示例代码丢进去，在编译模式中勾选“下次编译时模拟更新”，编译时就可以看到效果了。

App({

onLaunch: function(){

const updateManager = wx.getUpdateManager()

updateManager.onCheckForUpdate(function (res) {

// 请求完新版本信息的回调

console.log(res.hasUpdate)

})

updateManager.onUpdateReady(function () {

wx.showModal({

title: '更新提示',

content: '新版本已经准备好，是否重启应用？',

success: function (res) {

if (res.confirm) {

// 新的版本已经下载好，调用 applyUpdate 应用新版本并重启

updateManager.applyUpdate()

}

}

})

})

updateManager.onUpdateFailed(function () {

// 新版本下载失败

})

}

})

**上传代码的时候要带上版本号，然后微信会根据版本号判断是否为新版本**

**链接**<https://www.jianshu.com/p/9d679e5f8072>

