# 安装构建npm

## 1.1安装npm

**本例以Vant Weapp 为例（https://vant-ui.github.io/vant-weapp/#/home）**

在小程序 package.json 所在的目录中执行命令安装 npm 包：

在视图—终端打开一个终端，如下图所示：

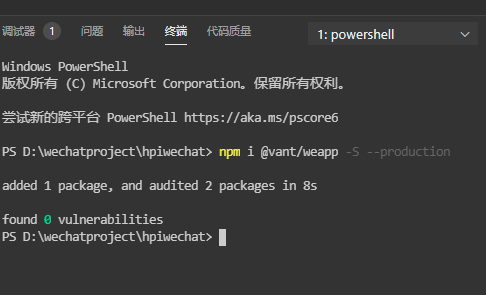
然后npm安装

安装之前，最好初始化一个包管理工具package.json

npm init –y

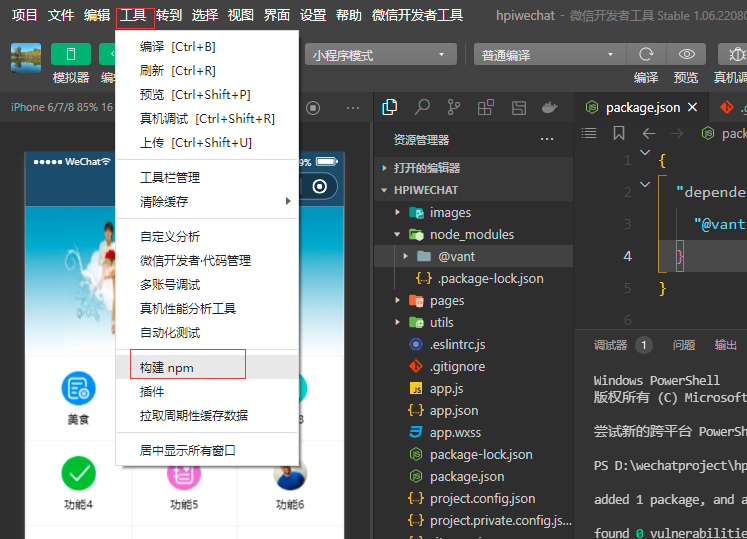
但是经过测试，初次npm包的时候，会自动生成包管理工具package.json

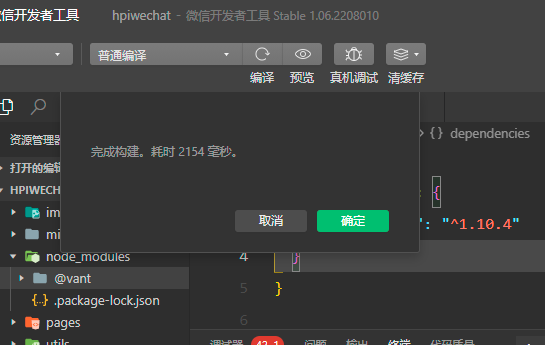
D:\wechatproject\hpiwechat> npm i @vant/weapp -S –production



## 1.2构建npm

点击开发者工具中的菜单栏：工具 --> 构建 npm

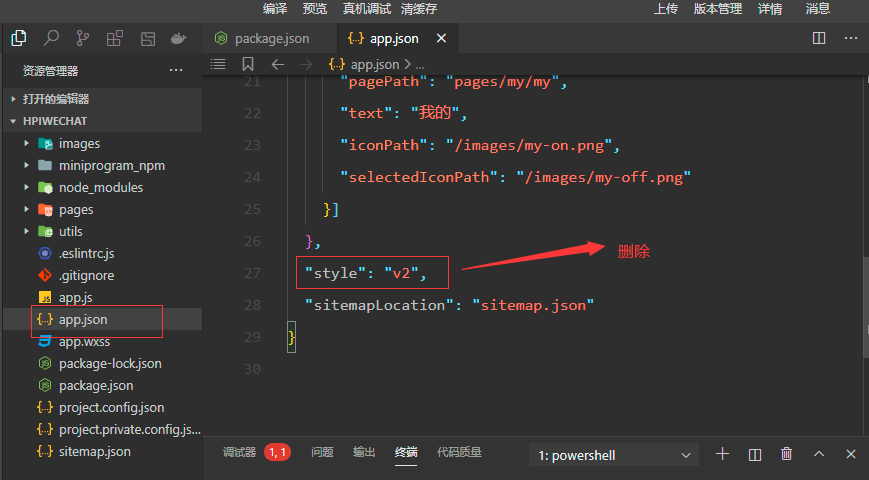




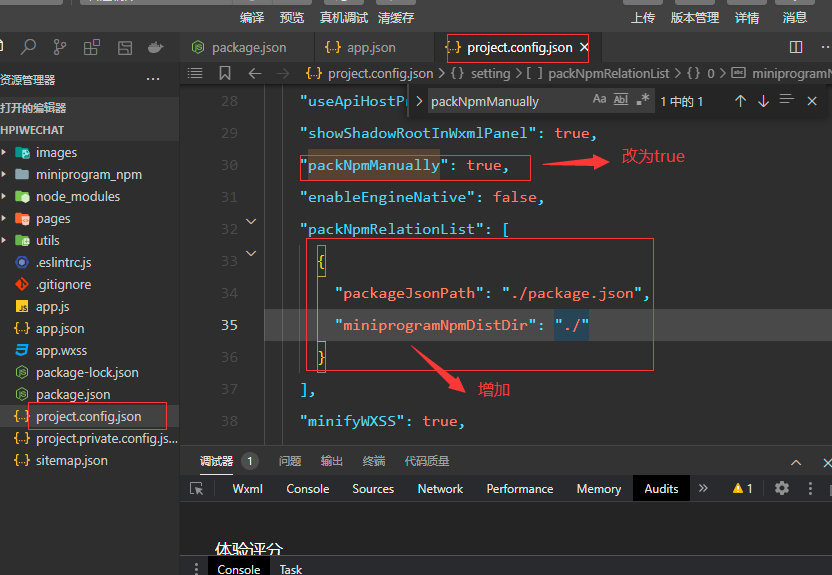
构建完成后，在详情-本地设置-使用npm模块打勾（新的基础库已经不需要此配置）

## 1.3修改app.json（初次npm需修改）

找到跟目录下app.json, 将 app.json 中的 **"style": "v2"** 去除，小程序的[新版基础组件](https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html" \l "style" \t "_blank)强行加上了许多样式，难以覆盖，不关闭将造成部分组件样式混乱。



## 1.4修改project.config.json(初次npm时修改)



注意配置项：

miniprogramNpmDistDir js下由于没有miniprogram目录，所以配置为./；ts下由于有iniprogram目录，需配置为："miniprogramNpmDistDir": "./miniprogram/"，如下图：



## 1.5 typescript 支持

如果你使用 typescript 开发小程序，还需要做如下操作，以获得顺畅的开发体验。安装 miniprogram-api-typings

# 通过 npm 安装

npm i -D miniprogram-api-typings

在 tsconfig.json 中增加如下配置，以防止 tsc 编译报错。



{

...

"compilerOptions": {

...

"baseUrl": ".",

"types": ["miniprogram-api-typings"],

"paths": {

"@vant/weapp/\*": ["path/to/node\_modules/@vant/weapp/dist/\*"]

},

"lib": ["ES6"]

}

}

## 1.5工程中应用

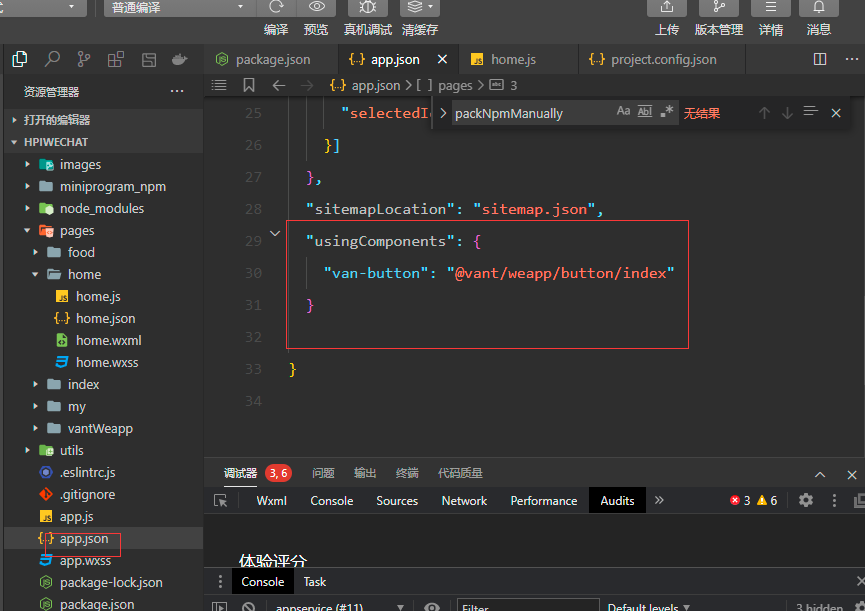
以 Button 组件为例，只需要在**app.json（全局引入）**或**index.json（局部引入）**中配置 Button 对应的路径即可。

所有组件文档中的引入路径均以 npm 安装为例，如果你是通过下载源代码的方式使用 @vant/weapp，请将路径修改为项目中 @vant/weapp 所在的目录。

1. 通过 npm 安装

如果是全局引入：在app.json 中加入如下：

"usingComponents": { "van-button": "@vant/weapp/button/index" }



另外，通过下载源码引入如下：



1. app.json

甘肃省灵台县什字镇胜利村新庄社085号