# Mpvue

# 一、目录结构

/---

|-->build //构建

| |-->...

|-->config //配置文件

| |-->dev.env.js //环境配置文件

| |-->index.js

| |-->prod.env.js

|-->dist //生成的微信小程序文件

|-->node\_modules //模块

|-->src //源码

| |-->components //vue组件

| |-->pages //页面

| |-->utils

| |-->app.json //配置路由

| |-->App.vue

| |-->main.js

|-->static //静态文件

|-->index //

|-->package-lock.json

|-->package.json

# 二、说明

1. 开发代码写在/src/pages/目录下
2. 添加页面需要从新运行npm run dev
3. 小程序路由配置信息写在app.json中
4. **需要nodejs、npm、mpvue、vue-cli支持！**
5. 关系映射

（1）main.js 导出的config字段，编译为小程序的\*.json文件，**config这个字段可以直接剪贴到\*.json文件中**

\*.vue文件的template部分编译为\*.wxml,

\*.vue文件的script部分编译为\*.js,

\*.vue文件的style部分编译为\*.wxss。

**/src/app.vue中script===>小程序app.js <colorui在这里引入>**

**/src/app.vue中style===>小程序app.wxss**

**/src/app.vue中temple===>小程序app.wxml**

1. src/main.js和src/App.vue会被编译为小程序的**app.js**、**app.json**、app.wxss。
2. src/components是公共组件部分，可以使用vue的组件重用机制
3. /src/App.vue中代码不能热更
4. **ESline格式错误不会显示在console中**
5. **修改/src目录下以外的文件需要手动运行项目**

# 开发记事

## 说明-使用流程

**npm install 后需要运行npm run dev后在小程序开发工具中打开，小程序开发工具才能识别**

小程序：app.json是小程序的全局配置文件，其包含了小程序的页面文件路径配置、窗口的全局样式信息、底部选项卡式菜单栏的配置，以及一些小程序网络超时的配置等等。

Mpvue：通过export default {config:....}实现，或在App.vue同级目录下新建App.json文件（直接将config后的对象填入）

小程序：app.js中包含了小程序的各种原生生命周期方法，如onLaunch、onShow等等。

Mpvue：App.vue实现等价的功能。（通常使用Vue的生命周期方法，但也兼容原生的生命周期方法）

-----------------------------------------------------------------------------------------------------------------

调用流程

/src/main.js 导入App.vue，将App.vue设置一个myType属性值

在page/下创建index.vue，main.js2个文件，由main.js调用index.vue，vue用于写小程序逻辑，main.js中可以使用export default {config:...} 导入小程序每一页的配置

-------------------------------

**微信小程序开发过程中要关掉验证域名**

## 1、安装流程

# 全局安装 vue-cli

$ npm install --global vue-cli

# 创建一个基于 mpvue-quickstart 模板的新项目

$ vue init mpvue/mpvue-quickstart my-project

# 安装依赖

$ cd my-project $ npm install

# 启动构建

$ npm run dev

-----------------------------------------------------------------------------------------------------------------

安装参数说明

**Project name**：项目名称 （最好与你的存放文件名一致）；

**wxmp appid**： 微信公众平台的appid  (开发者账号注册后由微信公众平台提供)；

**Project description**：项目描述（有默认，可直接跳过，也可根据自己项目描述）

**Author**：开发者信息（例如邮箱）

**Vue buid(Use allow keys)**：vue构建类型   （默认是运行runtime ，即不编译）

**Use VueX？**：是否使用VueX框架  （一般默认Y，VueX用于管理共享的数据，详细请查阅官方文档）

**Use ESLint to lint your code?**：是否启用ESLint代码风格  （相当严格的代码规范，出错则编译不通过，不过确实挺规范，习惯了就好，可以统一项目的代码风格，默认Y，不习惯到时可以在配置中关闭）

**Standard**：指JS的标准风格，即ESLint的代码规则  （Enter默认即可）

## 使用CSS

## 配置统一接口

使用config文件配置接口，方便之后修改稿

1、开发环境的API

**/config/dev.env.js**

module.exports = merge(prodEnv, {

NODE\_ENV: '"development"',

API\_BASE\_URL: '"http://127.0.0.1:8080/api"'

})

-------------

2、生产环境的API

**/config/prod.env.js**

module.exports = {

NODE\_ENV: '"production"',

API\_BASE\_URL: '"https://www.my-domain.com/api"'

}

--------------------

3、调用：  
const baseURL = dev.env.API\_BASE\_URL

const baseURL = process.env.API\_BASE\_URL

wx.request({

url: `${baseURL}/products`

})

## 使用组件

1. 在components创建one.vue文件，内包含模板，js，css
2. import one from "@/components/one"; //在要使用组件的文件导入
3. components: { one } //声明注册
4. 在模板中使用组件<one></one>
5. 重新运行 npm run dev
6. 关闭开启Esline

## 5、/config/index.js

module.exports = {

dev:{

useEslint: true, //注意是在dev对象下

}

}

## import x from ‘@/index’

@是在/bulid/webpack.base.conf.js文件里是这样配置的，可以这样理解：@/aa/bb === src/aa/bb ,把 @ 理解成一个变量，目前给 @ 赋的值是 src。

## 使用colorUI

复制colorUI/demo/colorui到/src/目录下与/pages同级

/src/main.js添加代码

import './colorui/animation.wxss'

import './colorui/main.wxss'

import './colorui/icon.wxss'

然后就可以复制demo中的代码进行开发，css库