VUE开发demo

作者：张涛

1. Vue介绍
2. Vue安装环境
3. Vue项目创建
4. Vue文件夹介绍
5. Vue组件介绍
6. Vue请求封装
7. Vue跨域设置
8. Vue打包部署

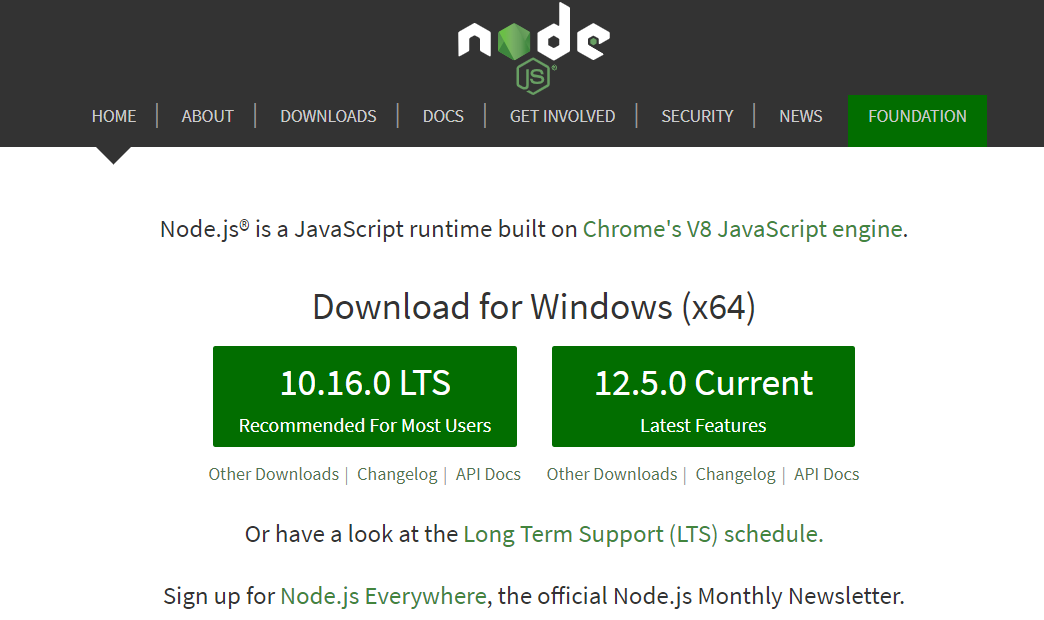
**Vue介绍**

Vue (读音 /vjuː/，类似于 view) 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

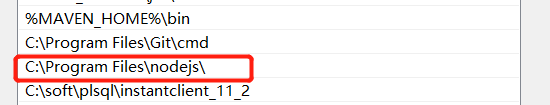
**Vue安装环境**

安装vue之前，确定你的电脑先安装了node.js，因为vue的创建需要node.js作为搭载的运行环境。

下载node.js **[https://nodejs.org/en/](#_top)**

如图：

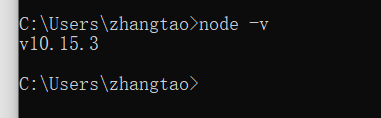
安装node.js，配置node本地环境变量.一般安装的过程中直接给加入到path中了。如下图

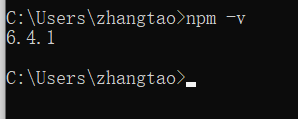


查看安装是否成功

Win+R打开cmd,输入node -v

如图：





要是安装不成功的话，好好反思一下自己。

**Vue项目创建**

1. 使用淘宝npm镜像

**npm  install  -g  cnpm  --registry=https://registry.npm.taobao.org**

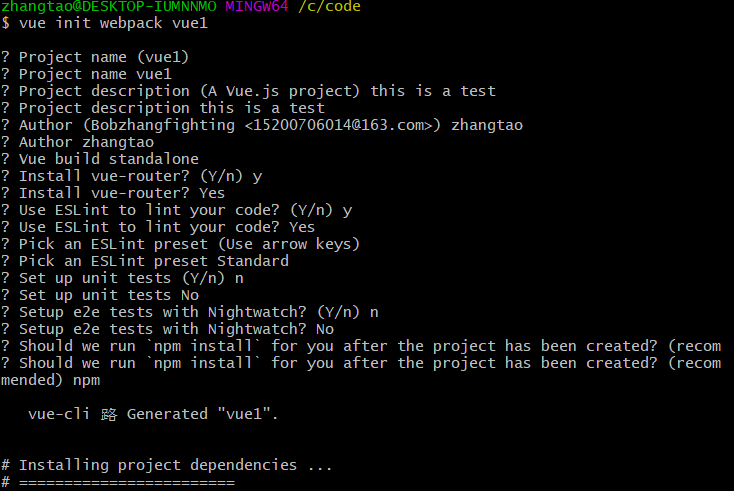
1. 初始化项目啊，安装vue-cli，当前vue-cli已经是3.0了

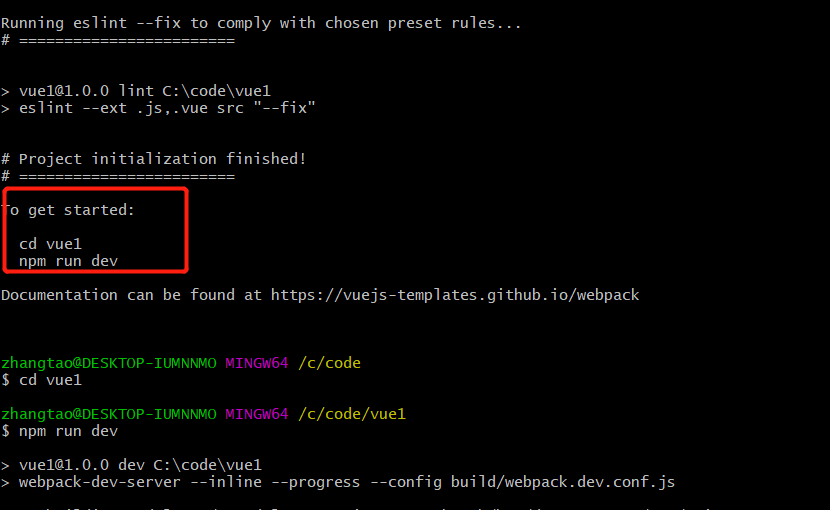
**cnpm install -g @vue/cli**

1. 创建项目

**vue create vue-test ---这是高级玩法**

**vue init webpack vue1 ---一般玩法**





1. 运行项目

按照上面的提示：

**cd vue1**（项目名）

**npm run dev** #启动

**创建项目命令总结:**

**npm install -g vue-cli**

**vue init webpack my-project**

**cd my-project**

**npm install**

**npm run dev**

**Vue文件夹介绍**

1. 如下图所示：
   1. ├── build/ # webpack配置文件
   2. │ └── ...
   3. ├── config/
   4. │ ├── index.js # 主要项目配置
   5. │ └── ...
   6. ├── src/
   7. │ ├── main.js # 应用入口文件
   8. │ ├── App.vue # 主应用程序组件
   9. │ ├── components/ # ui组件
   10. │ │ └── ...
   11. │ └── assets/ # 模块资源（由webpack处理）
   12. │ └── ...
   13. ├── static/ # 纯静态资源（直接复制）
   14. ├── test/
   15. │ └── unit/ # 单元测试
   16. │ │ ├── specs/ # 测试spec文件
   17. │ │ ├── index.js # 测试构建条目文件
   18. │ │ └── karma.conf.js # 测试跑步者配置文件
   19. │ └── e2e/ # e2e测试
   20. │ │ ├── specs/ # 测试spec文件
   21. │ │ ├── custom-assertions/ # e2e测试的自定义断言
   22. │ │ ├── runner.js # 测试跑步脚本
   23. │ │ └── nightwatch.conf.js # 测试跑步者配置文件
   24. ├── .babelrc # babel 配置
   25. ├── .postcssrc.js # postcss 配置
   26. ├── .eslintrc.js # eslint 配置
   27. ├── .editorconfig # editor 配置
   28. ├── index.html # index.html模板
   29. └── package.json # 构建脚本和依赖关系

**Vue组件介绍**

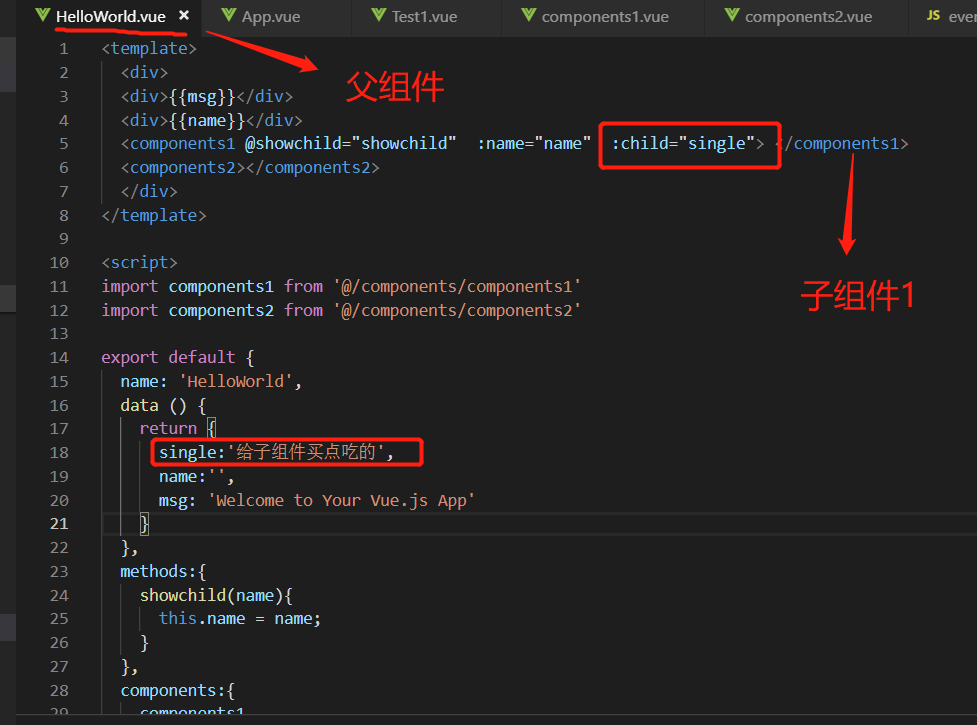
1. 什么是组件概念？

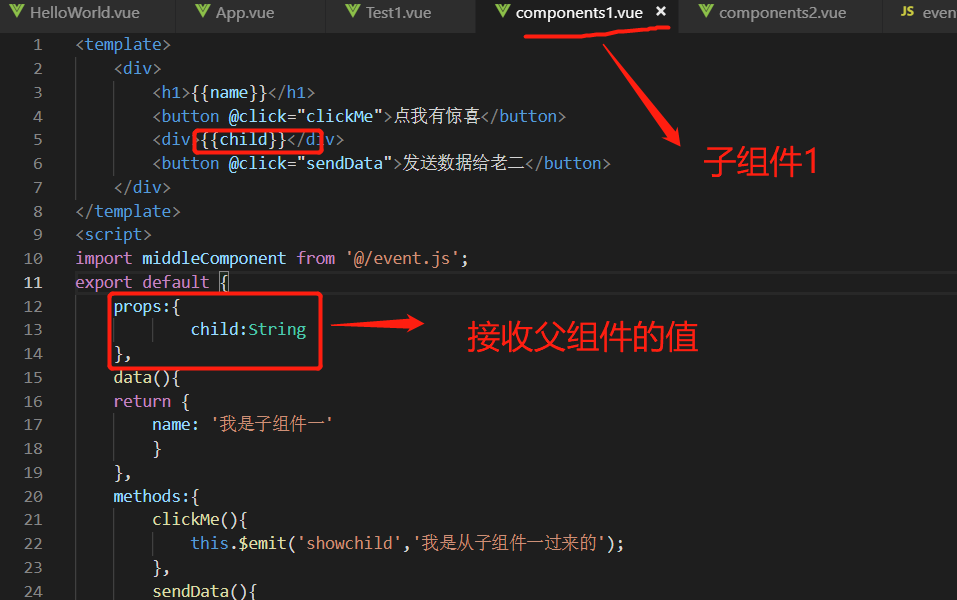
组件系统是Vue中的另一个重要的概念，因为它是一种抽象，允许我们使用小型，自包含和常可复用的组件构建大型应用，仔细想想，几乎任意类型的应用界面都可以抽象为一个组件树：那么到底组件是个什么东西呢？用一句话来进行简短的描述：

**组件就是可以扩展HTML元素，封装可重用的HTML代码，可以将组件看作自定义的HTML元素**。

1. 组件之间的通信(重点)

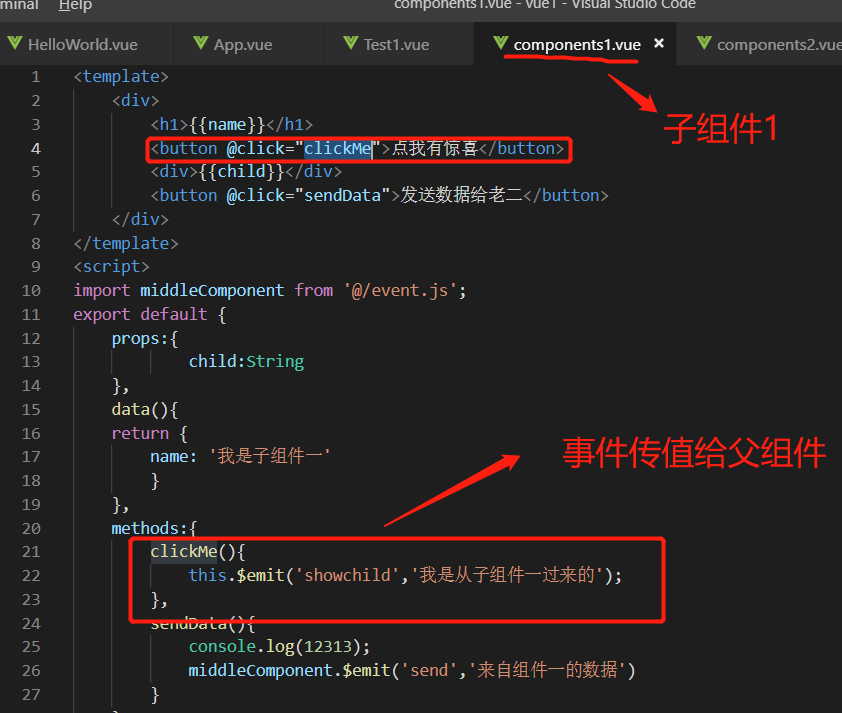
* 父组件传值给子组件

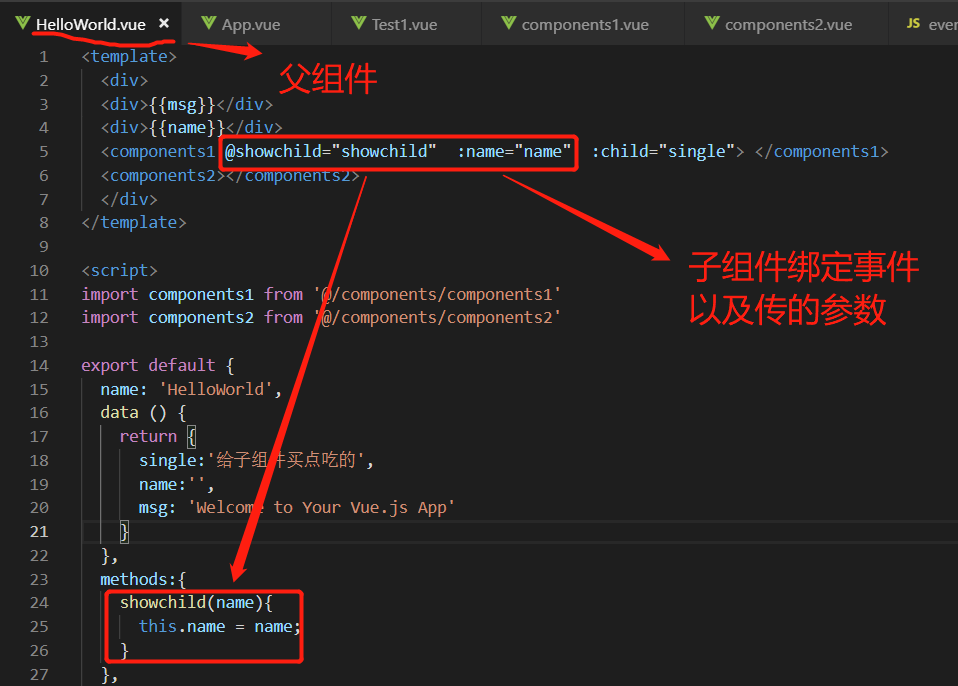




**父传值给子常规操作：在父组件中的子组件中绑定属性，子组件通过props来接收**

* 子组件传值给父组件





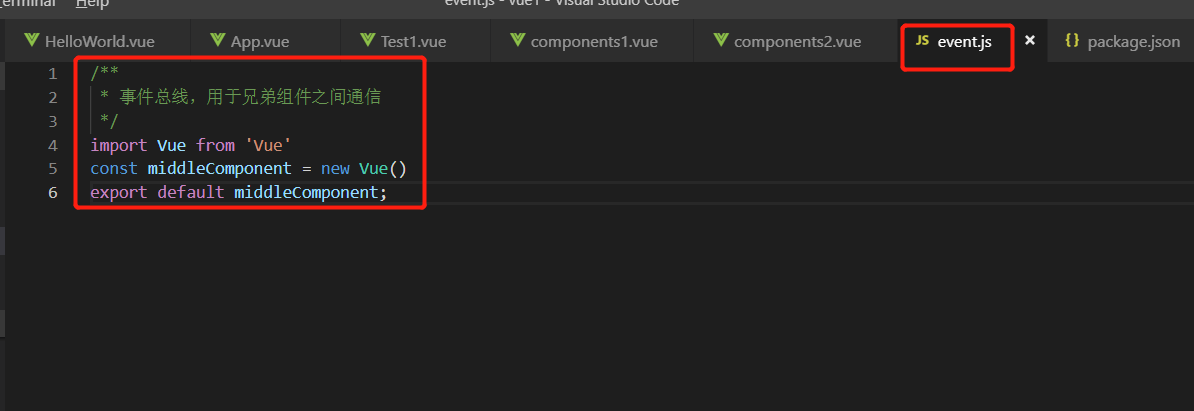
子传父常规操作：子组件通过this.$emit(‘事件’,参数); 父组件中绑定子组件的事件函数接收。

* 兄弟组件之间通信

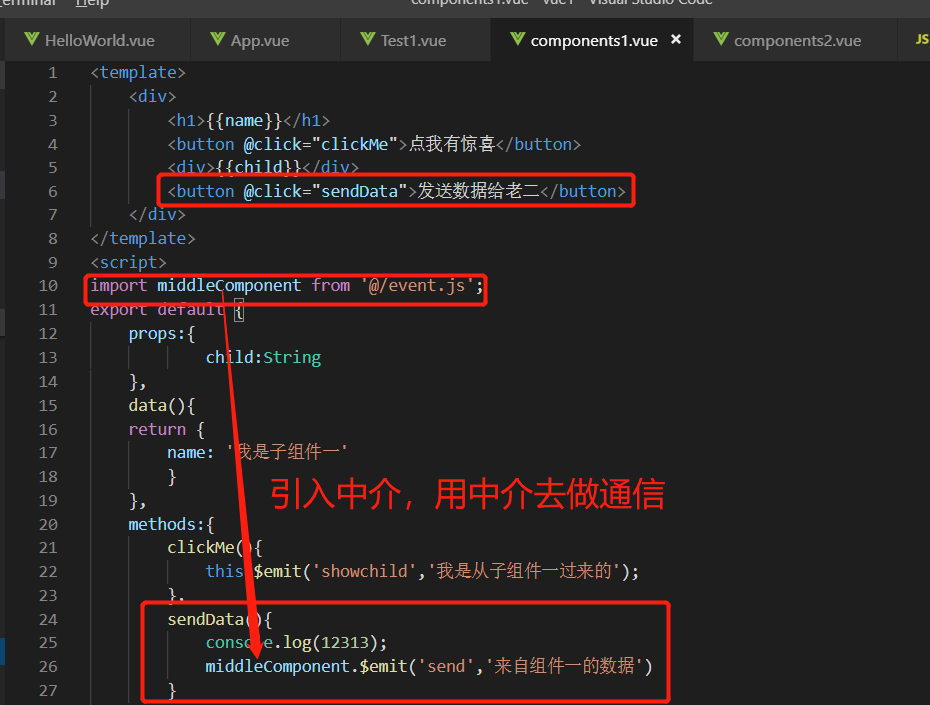
Vue开发中兄弟之间通信是必不可少的，那么怎么才能进行通信呢，找一个中介来处理。

即事件总线函数

**中介：**



组件1与组件2是兄弟组件：



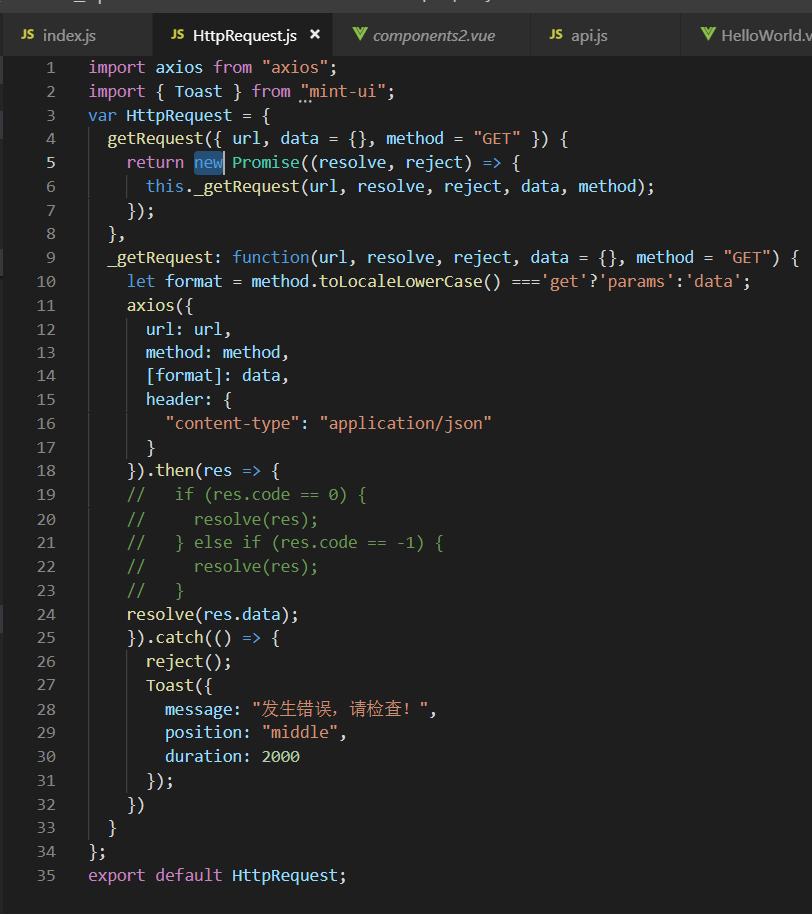
**组件1发送数据**：基本操作 中介.$emit(‘事件函数’，’参数’)

**组件2接收数据**：基本操作 中介.$on(‘事件函数’，’参数’)

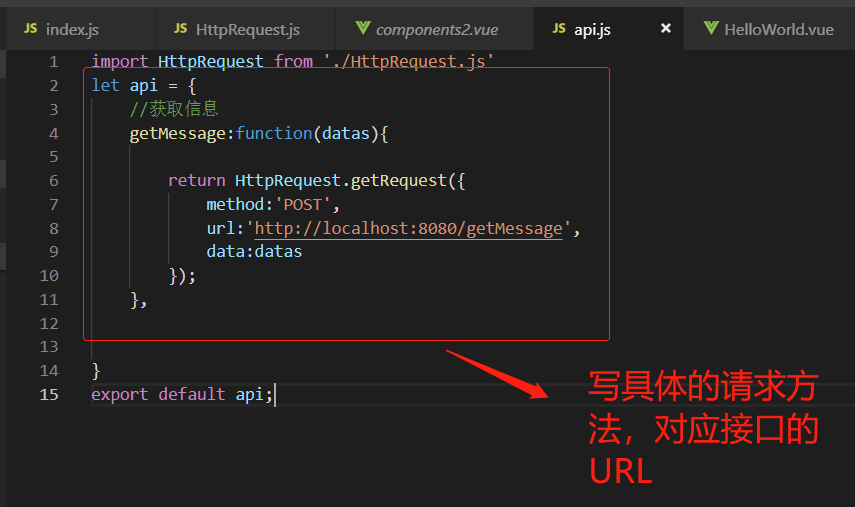


**Vue请求封装**

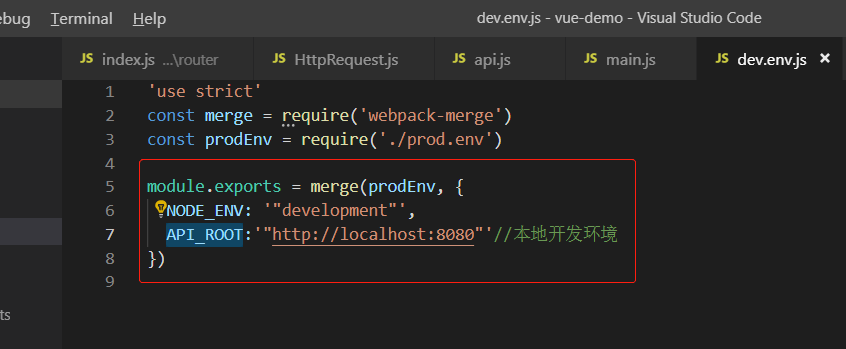
1、目前vue中封装请求一般是用Axios，Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。[https://www.kancloud.cn/yunye/axios/234845](#_top)

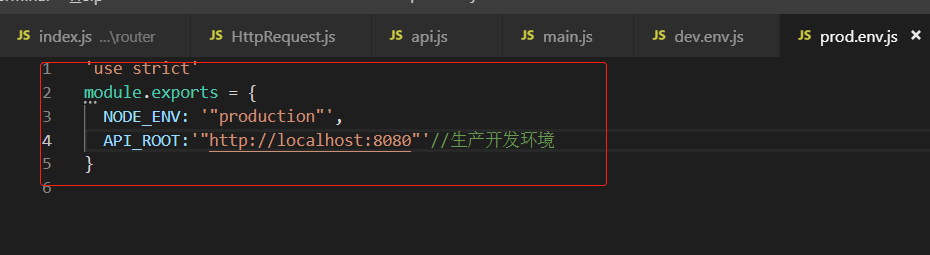


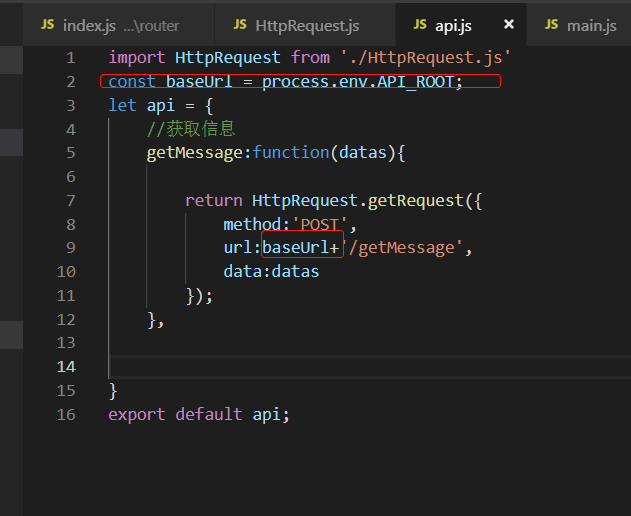
2、封装统一请求地址



1. 打包自动匹配请求前缀。





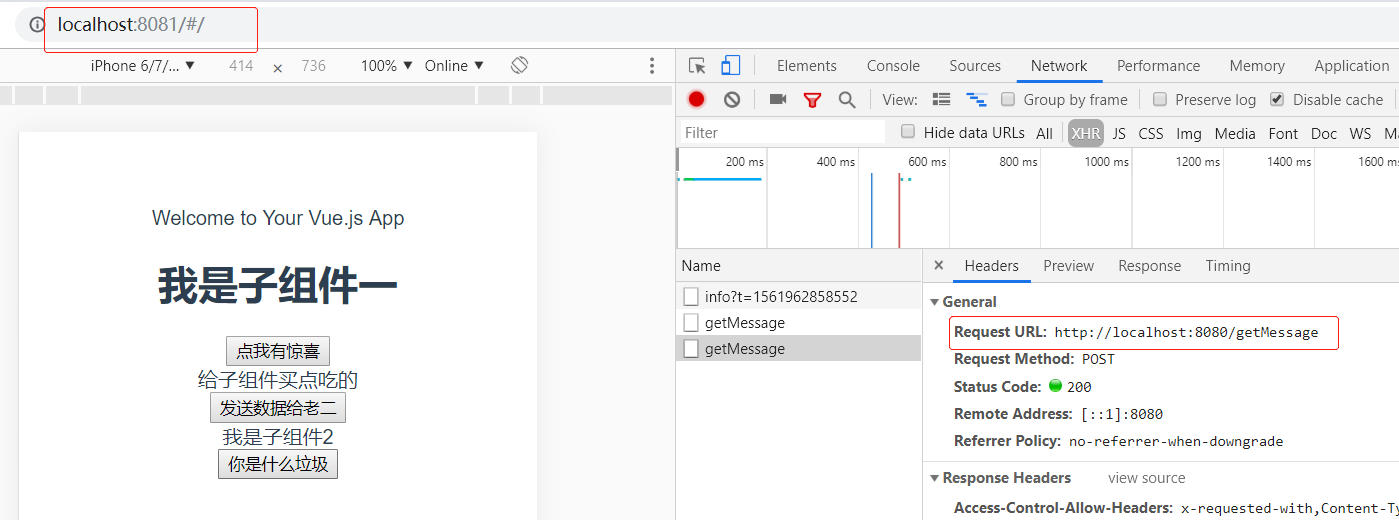


**Vue跨域设置**

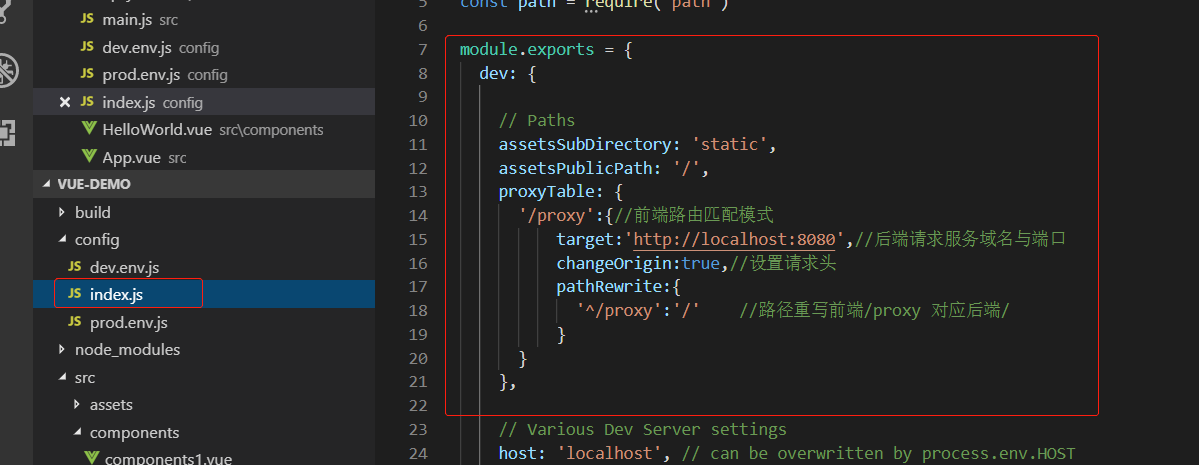
1、vue项目开发一般都是前后端分离的模式，首先来说一下后端跨域，如下图：

package com.java8.demo3.components;  
  
  
import lombok.extern.slf4j.Slf4j;  
import org.springframework.http.HttpStatus;  
import org.springframework.stereotype.Component;  
  
import javax.servlet.\*;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
*/\*\*  
 \** ***@Author*** *zhangtao  
 \** ***@Description:*** *前后端分离后端配置跨域  
 \** ***@Param:*** *\** ***@Return:*** *\** ***@Create:*** *2019/5/23 15:10  
 \** ***@version*** *v1.0  
 \*/*@Slf4j  
@Component  
public class CorsFilter implements Filter {  
  
 @Override  
 public void doFilter(ServletRequest req, ServletResponse res, FilterChain filterChain) throws IOException, ServletException {  
 HttpServletResponse response = (HttpServletResponse) res;  
 response.setHeader("Access-Control-Allow-Origin", "\*");  
 response.setHeader("Access-Control-Allow-Methods", "POST, GET, OPTIONS, DELETE");  
 response.setHeader("Access-Control-Max-Age", "3600");  
 response.setHeader("Access-Control-Allow-Headers", "x-requested-with,Content-Type");  
 HttpServletRequest request = (HttpServletRequest) req;  
 if (request.getMethod().equals("OPTIONS")) {  
 response.setStatus(HttpStatus.*OK*.value());  
 response.getWriter().write("OPTIONS returns OK");  
 return;  
 }  
 *log*.info("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*过滤器被使用\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 filterChain.doFilter(req, res);  
 }  
 @Override  
 public void destroy() {  
  
 }  
 @Override  
 public void init(FilterConfig filterConfig) throws ServletException {  
  
 }  
}

把上面的类丢都Java中即可，效果如下图



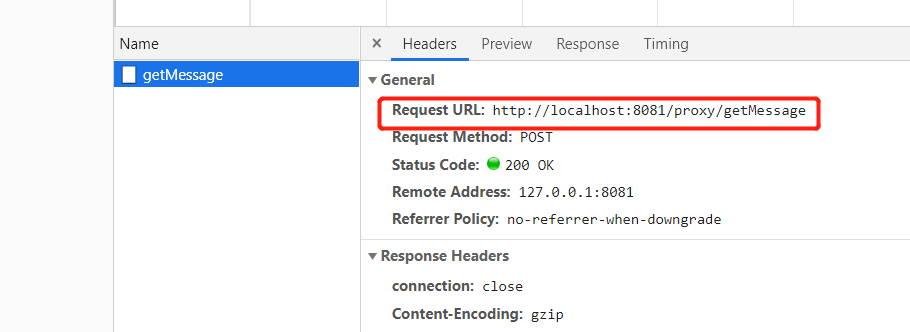
2、前端跨域，需要借助axios实现跨域



改一下api.js

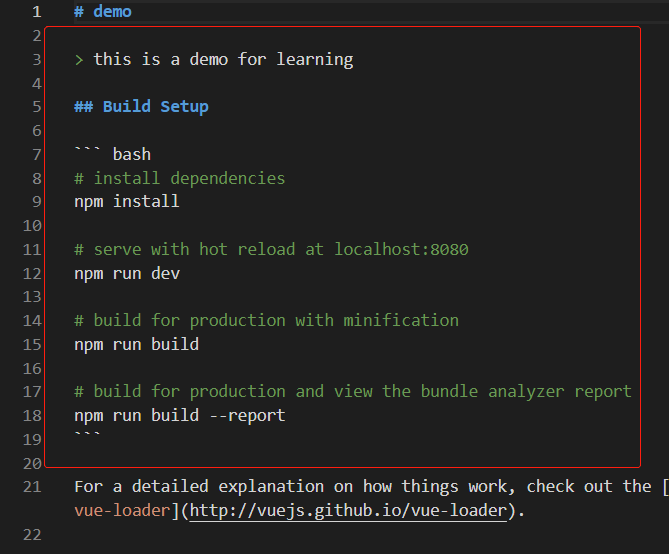


前端跨域的代理模式效果图如下：



**Vue打包部署**

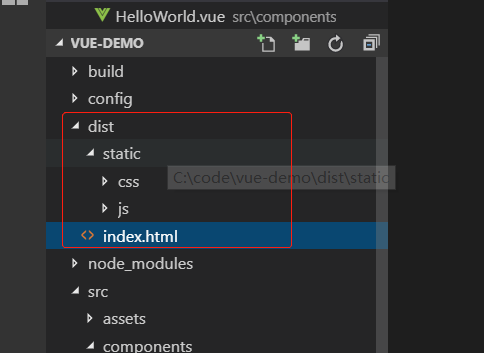
1、vue项目打包很简单的嘛，基本操作命令。首先看看你的项目的README.md文件里面写的啥玩意？



测试打包：npm run build

生产打包： npm run build --report

打包成功将会生成一个dist文件夹在项目里面



打包完成之后成了一个静态文件，至于怎么部署，犹如tomcat部署web项目一个样。