

# 主要内容

1. 项目的启动运行演示及说明
2. 项目描述
3. 项目技术架构(选型)
4. 项目前后台分离与mock数据
5. 接口的理解,测试和代理
6. 项目的模块化,组件化和工程化
7. vue项目状态管理利器vuex

# 项目的启动运行演示及说明

## 确保安装了node环境

1. 查看是否已经安装: node -v
2. 如果没有安装: 可以根据尚硅谷node课程视频去操作:
3. 地址: http://www.atguigu.com/html5\_video.shtml#nodejs

## 确保安装了mongodb, 并启动了对应的服务

1. 查看是否安装并启动了服务: 右键-->任务管理器-->服务-->MongoDB
2. 如果没有安装: 可以根据尚硅谷mongodb课程视频教程操作:
3. 地址: http://www.atguigu.com/html5\_video.shtml#nodejs

## 启动后台应用

1. 进入后台项目(gshop-sever\_final)根目录
2. npm start

## 启动前台应用并访问

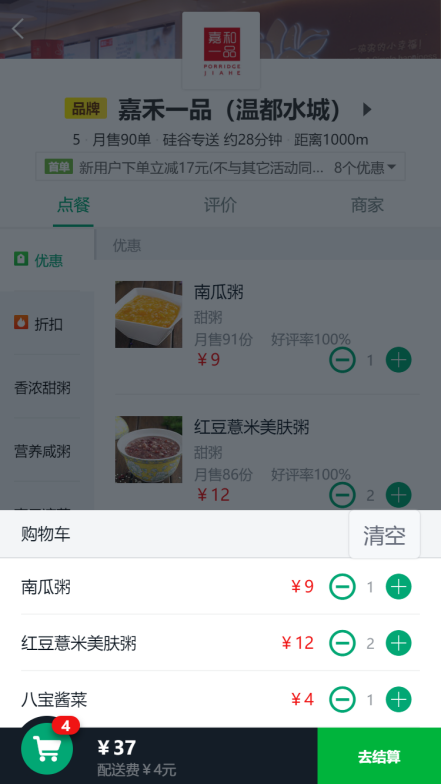
1. 进入前台项目(gshop-client\_final)根目录
2. npm start
3. 访问: <http://localhost:3000>

## 项目功能

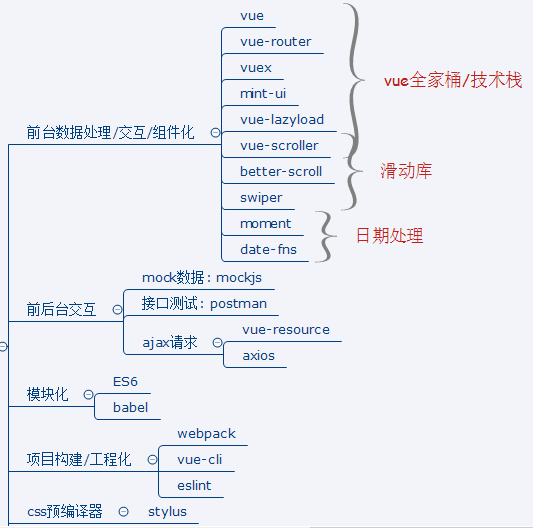
  

# 项目描述

1. 此项目是基于外卖的**前后台分离SPA**, 包含**前台应用和后台应用**
2. 包括**商家**, **商品**, **购物车**, **用户**等多个子模块
3. 使用**Vue全家桶**+**ES6**+**Webpack**等前端最新最热的技术
4. 采用**模块化**、**组件化**、**工程化**的模式开发

# 项目技术架构(选型)



# 项目前后台分离与mock数据

## 项目前后台分离

1. 包括2个项目, 一个后台项目, 一个前台项目
2. 后台项目: 向前台提供API接口, 只负责数据的提供和计算，而完全不处理展现
3. 前台项目: 通过Http(Ajax)请求获取数据,　在浏览器端动态构建界面显示数据

## mock数据/接口

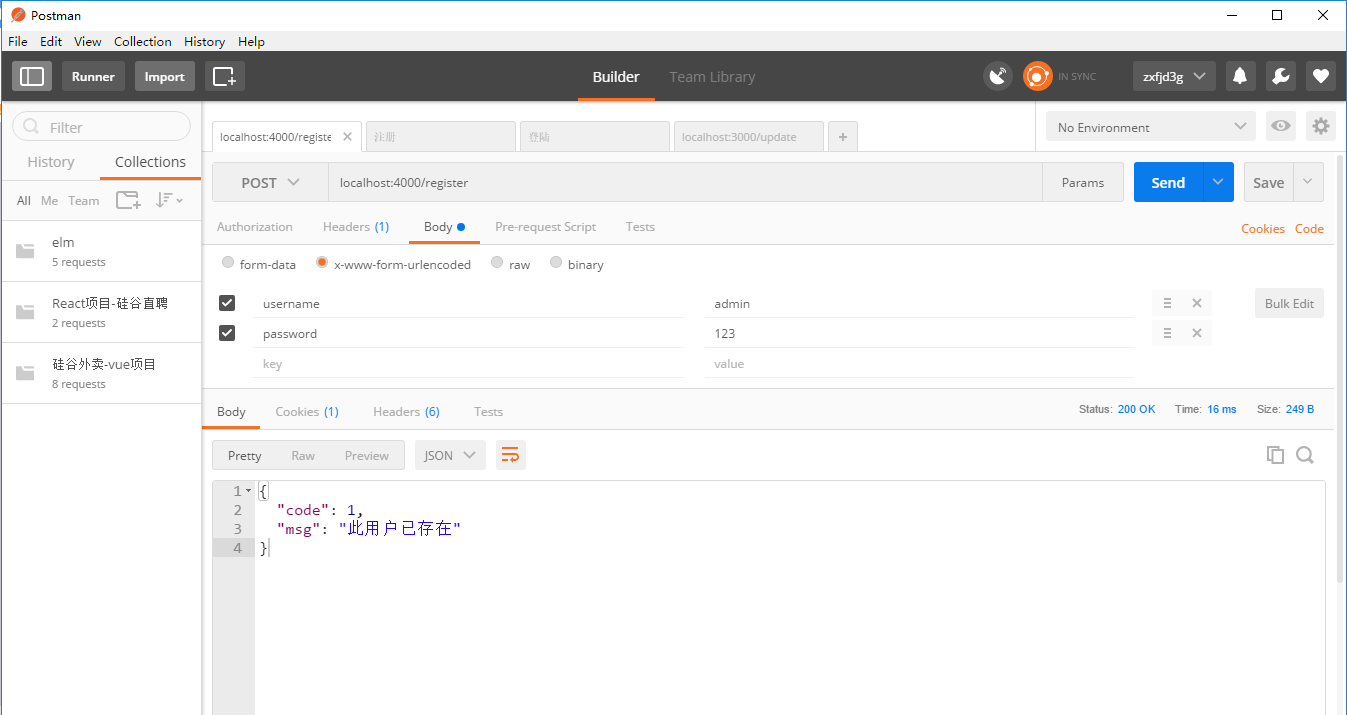
1. 对于前后台分离的前台项目来说, 如果后台还没有开发好, 可以自己来mock数据接口
2. 使用mockjs库可以轻松的实现mock数据/接口
3. 官网: <http://mockjs.com/>

# 接口的理解,测试和代理

## 接口的理解

1. url
2. 请求方式
3. 请求参数格式
4. 响应数据格式

## 接口测试(使用postman)



## 接口代理

1. 理解代理:
   1. 代理是运行在前台应用所在的服务器上的程序, 用来解决前后台的跨域ajax请求
   2. 它拦截前台应用的ajax请求, 并转发请求到指定的目标(后台应用)地址
   3. 得到的响应数据后自动交给前台应用, 从而实现了跨域请求, 但浏览器看到的不是
2. 配置代理: config/index.js

|  |
| --- |
| **proxyTable**: {  **'/api'**: { *// 匹配所有以 '/api'开头的请求路径* **target**: **'http://localhost:4000'**, *// 代理目标的基础路径* **changeOrigin**: **true**, *// 支持跨域* **pathRewrite**: {*// 重写路径: 去掉路径中开头的'/api'* **'^/api'**: **''** }  } } |

# 项目的模块化,组件化和工程化

## 模块

1. 理解: 向外提供特定功能的js程序, 一般就是一个js文件
2. 为什么: js代码更多更复杂
3. 作用: 复用js, 简化js的编写, 提高js运行效率

## 组件

1. 理解: 用来实现特定(局部)界面功能效果的代码集合(html/css/js/image)
2. 为什么: 一个界面的功能更复杂
3. 作用: 复用编码, 简化项目编码, 提高运行效率

## 模块化

当应用的js都以模块来编写的, 这个应用就是一个模块化的应用

## 组件化

当应用是以多组件的方式实现, 这个应用就是一个组件化的应用

# vue项目组件状态管理利器vuex

## vuex是什么?

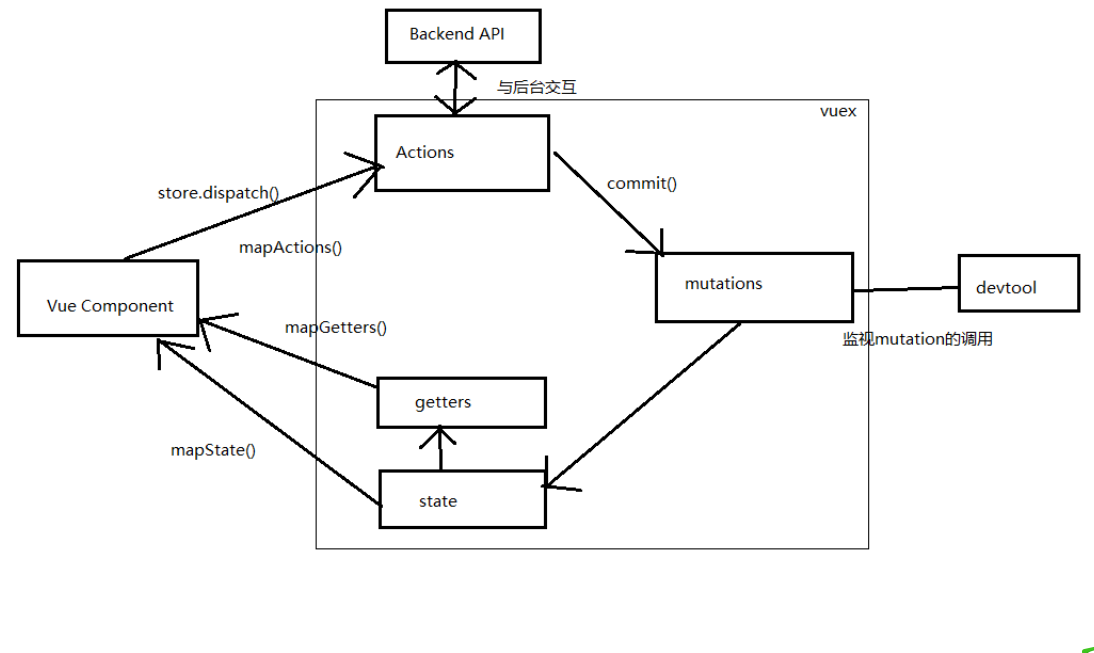
1. vue的插件库
2. 用于管理vue项目中多组件共享的状态数据
3. 补充: 一般从后台获取的数据都会交给vuex管理

## 多组件共享状态的问题

1. 多个视图依赖于同一状态
2. 来自不同视图的行为需要变更同一状态
3. 以前的解决办法
   1. 将数据以及操作数据的行为都定义在父组件
   2. 将数据以及操作数据的行为传递给需要的各个子组件(有可能需要多级传递)
4. vuex就是用来解决这个问题的

## vuex的结构





## 基本编程分析