# 核心需求：

用户端——展示商品、购物车、下单、支付、订单、用户

后台管理——商品管理、品类管理、订单管理、管理员登录

## 用户端：

商　品-->首页、商品列表、商品详情

购物车-->购物车数量、添加删除商品、购物车提交

订　单-->订单确认（地址管理）、订单提交、订单列表、订单详情

支　付-->支付

用　户-->登陆、注册、个人信息、找回密码、修改密码

## 管理后台：

商品管理-->添加/编辑商品、查看商品、下架

品类管理-->添加品类、查看品类

订单管理-->订单列表、订单详情、发货

权　　限-->管理员登录

# 架构设计

## 分层架构

定义：把功能相似，抽象级别相近的实现进行分层隔离

优势：松散耦合（易维护、易复用、易扩展）

常见分层方式：MVC、MVVM

## 模块化

定义：解决一个复杂问题时，自顶向下逐层把系统划分成若干模块的过程

意义：解耦、可并行开发

模块化方案：AMD、CMD、CommonJS、ES6

## 架构设计

# 技术选型

软件过程、前后端分离、构建工具、框架、版本控制、发布方式

## 软件过程选择——敏捷开发

定义：以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发

是一种迭代的意识和方法，而不是概念和工具

优点：能够应对满足不断变化的需求

不足：对团队成员的能力要求比较高

## 前后端分离方式——完全分离2

* + 1. 不分离：

前后端公用一项目目录，甚至页面内嵌js、css

本地开发环境搭建成本高

共同维护成本高

发布风险高

* + 1. 部分分离：

后端负责页面模板（JSP/Velocity/Freemarker）

本地开发环境搭建成本较高

更新页面模板仍需后端协助，效率不够高

需要前后端同时发布

* + 1. 完全分离1

velocity，发布的时候同步到后端

优点：完全分离，能直接生成动态的模板，利于SEO

缺点：系统复杂度高、需要前后端同时发布

* + 1. 完全分离2

纯静态html、完全通过接口做数据交互

优点：完全脱离后端模板，系统复杂度低

缺点：不太利于SEO

优化方案：Server Render / 蜘蛛定制页面

## 框架选型

框架：JQuery（严格来说不是框架）、Angular、Avalon、Vue、React

样式框架：CSS、Less、Sass

* + 1. 用户端选择：

要求：求稳、用户类型多样、有SEO要求，多页应用

选择：JQuery、原生CSS

* + 1. 管理系统

要求：求快、用户类型单一、无SEO要求

选择：React、Sass

## 构建工具——Webpack

Grunt、Gulp、Webpack

## 版本控制——Git

SVN、Git

## 发布方式

* + 1. 发布过程

拉取待发布代码 🡪 编译打包 🡪 发布到线上机器

* + 1. 域名发布

html 🡪 www. / admin.

js + css 🡪 s.

image 🡪 img.

# 开发环境搭建

## 语言环境

用户端：jQuery Webpack(基于Node.js) Git

## 开发工具

开发工具：VS Code

浏览器：Chrome

代理工具：Charles/Fiddler

## node.js、npm的介绍及安装

* + 1. node.js简介

定　　义：JS的服务端运行环境

用　　途：构建工具Webpack的环境依赖等

特　　点：单线程、异步编程

应用场景：低运算、高I/O

* + 1. node.js安装

官网地址：https://nodejs.org/

Mac：node-vxxx.pkg

Windows：node-vxxx.msi

Linux：node-vxxx-linux-x64.tar.gz

## Git介绍及安装

* + 1. 简介

Git是一款免费、开源分布式版本系控制系统，用于敏捷高效地处理任何或小或大的项目

* + 1. Git安装

官网地址：https://git-scm.com/

Mac：https://git-scm.com/download/mac

Windows：https://git-scm.com/download/win

* + 1. Git常用命令

git init —— 在当前目录新建一个Git代码库

git clone —— 下载一个项目和它的整个代码历史

git status —— 显示有变更的文件

git add . —— 添加当前目录的所有文件到暂存区

git commit -m [message] —— 提交暂存区到仓库区

git pull —— 取回远程仓库的变化，并与本地分支合并

git push —— 上传本地指定分支到远程仓库

git branch —— 列出所有本地分支

git branch -a —— 列出所有本地分支和远程分支

git checkout [branch-name] —— 切换到指定分支，并更新工作区

git merge [branch] —— 合并指定分支到当前分支

## VS cod 安装

官网地址：https://code.visualstudio.com/

插件推荐：Auto Close Tag、Auto Rename Tag、ESLint、Emmet、Debugger for Chrome、HTML Snippets等

# 项目初始化与基于模块化的脚手架搭建

## 目录结构的设计



## 项目脚手架搭建概要

* + 1. npm安装依赖包

安装依赖包：npm intall [xxx@v.v.v](mailto:xxx@v.v.v)

卸载依赖包：npm uninstall [xxx@v.v.v](mailto:xxx@v.v.v)

参数：-g （全局安装）

参数：--registry=http://registry.npm.taobao.org （将npm源指向淘宝npm镜像）

* + 1. Webpack

设计思想：require anything

加载方式：各种loader插件

编译方式：commonjs模块 🡪 function类型模块

官方文档：<http://webpack.github.io/docs/>

* + - 1. Webpack安装

Webpack需要全局安装：> npm install webpack -g

本地项目安装1.15.0版的webpack：> npm install webpack@1.15.0 --save-dev

通过 >webpack -v 查看webpack版本号

* + - 1. Webpack安装的疑问

【问】为什么全局安装的时候不用版本号？

【答】涉及到npm的加载原理，npm会优先使用项目本地npm安装包，如果找不打则会在全局的npm安装包里面找，全局安装就是为了使用webpack的命令，真正使用的则是本地npm的版本安装包

【问】为什么用1.15.0版本，官网不是建议升级2.x了吗？

【答】webpack降级到1.x版本是因为2.x版本压缩js的uglify会在代码中生成一行代码叫Object.default，default是js switch case的关键字，IE8看到这个会不兼容，所以要降回1.15版本

【问】问什么在安装后面加 –save-dev？

【答】这是npm的另外一种机制，会把包的信息存放到package.json中，记录项目的依赖。