

软件体系结构

课程报告

任课教师：张贺

姓名：丁旭行 姓名：周晋锴

学号：502022320003 学号：522022320220

姓名：杨雨辰 姓名：周垭洁

学号：522022320178 学号：522022320224

2023 年 6 月 6 日

开源项目分析

开源项目：CRMEB

链接：https://gitee.com/ZhongBangKeJi/crmeb_java

star 数量：4.3k

文档：https://doc.crmeb.com/java/crmeb_java

项目简介

CRMEB Java 版商城系统是一款基于 Java + Uni-app 开发的新零售社交电商系统，能够真正帮助企业基于微信公众号、小程序、移动端等，实现会员管理、数据分析、精准营销的电子商务管理系统。

项目特点

系统采用 SpringBoot 框架开发，基于业界主流技术，代码有详细注释和完整手册。系统前后端分离，前端 PC 管理端使用 Vue 和 Element UI 框架开发，移动端使用 UniApp 框架。使用标准的 RESTful 接口和数据格式，有更清晰的逻辑层次和更高的 API 复用性。支持 Redis 队列避免流量高峰，解耦合，高可用。无缝的事件机制，方便扩展新行为和二次开发。提供数据导出功能,方便个性化分析。使用 ECharts 实现用户、产品、订单和资金等数据统计分析。完善的权限管理,后台有多种角色和多重身份,权限控制到按钮级别。利用 Vue 表单生成器，通过拖拽配置表单，减少前端重复工作，提高开发效率。

项目功能列表

序号	模块	功能介绍
1	商品	能够对商品的状态分类管理（出售中、待上架、库存中、已售罄、库存警戒、回收站）、添加产品、添加商品分类等功能
2	会员	站内会员的管理（发放优惠券、发通知、发图文消息、增加余额、会员行为详情）、站内通知、微信端用户管理（微信用户的管理、分组、行为观察、添加标签）等功能
3	营销	能够管理优惠的发放和制作、用户积分的统计使用情况、秒杀产品的管理等
4	财务	能够对用户的消费、充值、返佣等进行记录
5	订单	能够完成用户的订单管理（发货、订单详情、修改订单、订单备注、订单记录、订单退款）、售后服务（评论的回复与删除）

6	分销	后台有分销统计管理，分销可以设置人人分销和指定人分销，也可以自己稍微开发一下修改相应规则
7	数据统计	图表统计分析（财务统计、产品统计、会员统计、营销统计、分销统计、交易统计等）
8	设置	能够完成管理员对网站的商品资料管理（添加父子类、商品添加、属性快速生成、商品审查）、商品交易（外理订单、发货查询）、会员管理（会员审查）、操作管理（管理员添加、管理员审查、管理员退出）、系统配置、后台通知等功能
9	内容	管理文章分类（添加分类、删除分类、修改分类）、管理文章
10	维护 q	查看系统日志、文件变动效验、刷新网站缓存、在线更新系统、清除数据、文件管理等功能
11	权限管理	可以创建不同身份，不同管理员，同一个管理员可以拥有多重身份，权限可以控制到每一个控制器函数

需求分析

1. 电商平台核心功能：用户、商品、订单。需要实现用户管理、商品分类与管理、订单流程等功能。

a. 用户管理：需要实现用户的注册、登录、资料管理等功能。用户是平台的基石，用户管理是核心需求。

- 支持手机号码、邮箱、社交账号注册/登录
- 用户可以编辑个人信息，上传头像，设置登录密码
- 需要对用户权限进行管理，区分管理员、店铺管理员和普通用户

b. 商品管理：需要提供完善的商品录入、分类、审核与上下架管理功能。这直接决定了平台的商品丰富度和购物体验。

- 支持对商品进行深度分类管理，包括一级分类、二级分类和三级分类
- 商品录入过程中需要设置商品名称、图片、规格、价格、描述等信息
- 需要支持对商品的审核、上架和下架操作，对商品的整个生命周期进行管理
- 需要对商品的门店信息、评价、咨询等相关内容进行管理

c. 订单管理：这是电商项目的经济核心，需要提供完整的下单、支付、配送、评价等功能。

- 下单过程需要收集用户的收货人信息，包括姓名、电话、收货地址等
- 需要支持在线支付和货到付款，与各大支付平台进行对接
- 需要支持物流信息的上传与查询，以便买家进行物流跟踪
- 交易完成后需要允许买家进行商品评价

2. 商品销售渠道：APP 端、小程序端、PC 端。需要实现多端同步的商品管理与展

示。

a. **APP 端展示:** APP 端以移动端为主, 需要考虑操作便捷性与界面友好性。

- 商品图片及详情需要适配手机屏幕, 控制在可操作的大小, 同时保证清晰度
- 商品列表及分类展示要简洁直观, 一屏内展示所有分类或最大数量的商品
- 需要提供流畅的滑动、放大、旋转等操作方式, 方便用户浏览商品详情
- 下单、支付等流程要尽量少键盘输入, 更多依赖选择及滑动操作
- 个人中心要突出用户关注的订单、收藏、足迹等信息, 同时保持简洁

b. **小程序端展示:** 小程序更侧重快速完成交易, 展示以秒杀、砍价为主, 注重推送更新信息。

- 启动页一屏内要突出秒杀、砍价等活动入口, 方便用户快速参与
- 商品详情页也需要突出活动信息, 让用户立即参与并下单
- 购物车及下单页面需要支持快速结算, 少数键入就能完成支付
- 需要提供更新提醒, 推送最新的商品促销、限时活动信息
- 个人中心要突出用户的优惠券、积分等, 提醒其尽快使用或兑换

c. **PC 端展示:** PC 端有更大展示空间, 需要提供丰富详尽的商品信息及分类, 注重详情页的美观度。

- 首页背景图及广告位等视觉要素更加重要, 起到吸引用户的作用
- 商品图片及详情内容都需要大图展示, 且加载要迅速流畅
- 商品筛选及分类展示要尽量全面详尽, 提供更多条件及更详细的子类
- 商品详情页在保证信息细节详尽的同时, 视觉元素也同样重要, 要求较高美观度
- 下单过程可以提供更多选项及输入方式, 不像移动端那么追求极简
- 个人中心的订单、收藏、评论等信息都需要最大化展示, 以丰富用户体验

3. **营销功能:** 优惠券、砍价、拼团、秒杀等营销手段。需要提供灵活的营销机制来提高用户购买率。营销功能是电商项目的核心诉求之一, 直接影响平台的交易量与收入。

a. **优惠券:** 这是一种常见的营销手段, 需要提供管理与使用功能。

- 需要支持多种优惠券类型设置: 满减券、折扣券、代金券等
- 需要设置有效期, 可针对部分商品或全场商品使用
- 发放渠道需要丰富: 注册赠券、购物赠券、分享得券等
- 使用规则需要详细: 是否与其他优惠共享、使用门槛、最大抵扣额等
- 发放与使用的数据统计与分析是关键, 需要实时跟踪并定期总结各类优惠券的效果

b. **砍价活动:** 这是提高商品销量与用户兴趣的途径之一, 砍价全过程需严密控

制。

- 需要支持商家选择参与砍价的商品，确定初始价格与底价
- 需要设置每次砍价的时间及金额间隔，控制砍价速度与幅度
- 需要实时更新砍价的进度与定价，同步到所有销售渠道
- 砍至底价时需要立即结束活动，并限制商品数量，先到先得
- 需要分析每次砍价引起的商品销量变化，得出最佳的砍价方案

c. 拼团活动：这一活动可以发动更多人一起购买，共享优惠，同时提高商品热度。

- 需要支持多种拼团模式：时间限制、人数限制、数量限制等
- 拼团成功与失败的条件及结果需要详细设置，包括优惠力度、限购等
- 商品的拼团价格通常更低，需要实时同步并与非拼团价格对比
- 需要提供拼团进度显示，方便用户邀请更多人一起参与活动
- 拼团结束需及时更新商品价格与限购信息，并分析活动的成功效果

d. 秒杀活动：这是利用时间限制来促成商品抢购与提高销量的有力手段。

- 需要严格控制秒杀开始与结束时间，商品数量要显著少于平常
- 秒杀价格需要远低于原价，以引发用户的抢购欲望
- 需要提供倒计时提示，让用户寻找最佳下手时机
- 活动开始时系统需要有较强的承受能力，避免高并发引起故障
- 结束后需要及时下架商品并总结活动效果

4. 支付结算：需要集成各大支付渠道，比如微信支付、支付宝等，并实现支付结算流程。

a. 支付渠道：需要对接各主流在线支付渠道，满足用户的多种支付习惯。

- 需要对接微信支付、支付宝、云闪付等渠道
- 需要对不同渠道的支付手续费进行配置与对比
- 支付渠道的选择需要提供给用户，并据此确认支付金额
- 渠道切换或增加新的渠道需要有严密的测试流程，确保支付成功率

b. 支付流程：这是从下单到支付成功的全过程，需要严密监控。

- 下单完成后需要直接进入支付页面，提供客户选取支付渠道
- 支付中需要实时轮询支付平台，获取支付状态与结果
- 支付成功后需更新订单状态，同时触发发货流程
- 支付失败需提示用户并提供重新支付入口，部分渠道需要继续轮询
- 全过程需要监控系统性能表现，特别是高并发下的支撑能力

c. 退款流程：这是实现客户权益与营销手段的必要功能。

- 退款申请需要提供退款原因的选择，及客户留言反馈
- 不同支付渠道的退款时间及要素需要配置明确，并提醒客户
- 退款成功与失败的条件判断及客户疑问解答内容需要准备
- 退款审核人员需要指定，并对不同金额设置备注或审批流程
- 退款后订单及支付信息需及时更新，以保证数据的完整性与准确性

d. 结算流程：这是平台获得收入的最后一步，涉及资金安全，更需要严密。

- 需要指定结算周期，如 T+1 日结算、T+7 日结算等
- 需要生成结算报表，包括订单号、交易时间、收入等信息
- 需要对结算金额进行复核，与订单收入统计数据进行核对
- 不同支付渠道的到账时间需要明确，并控制在一个结算周期内
- 资金转账至平台时需要及时对账，确保金额准确无误后更新数据

5. 统计分析：需要提供商品统计、销售额统计、流量统计等数据统计与分析功能，以支持商家定制营销策略。

a. 商品统计：需要对商品销量、评价、访客等数据进行跟踪，发掘商品潜力或问题。

- 需要统计每件商品的销量、退款率、访客数等数据
- 需要标识商品的畅销与爆款，也需关注销量低迷商品
- 需要根据评价内容分析商品存在的问题与改进点
- 不同分类或品牌的商品需横向对比，发掘用户偏好
- 新品上线后的表现也需要跟踪，调整市场策略或产品本身

b. 销售额统计：需要对交易额、客单价、支付比例等关键数据进行统计，判断销售额变化趋势及支付能力。

- 每日/周/月的交易总额与对比数据需实时跟踪
- 不同渠道及支付类型的销售额占比需持续统计
- 客单价的变化可以判断购买力及已购商品类别变化
- 退款比例的大小可以判断问题商品或客户服务质量
- 不同分类或品牌销售额的变化也需要关注，定期优化营销策略

c. 流量统计：需要对各销售渠道的访问量、访客数等数据进行跟踪，判断较为活跃的渠道或推广效果。

- 网站、小程序、APP 的日访问量与浏览深度需要统计
- 不同推广渠道访客的来源及转化率需要跟踪
- 新客与老客的访问频率及购买转化率需关注
- 访客的留存率需要统计，判断流失的可能性
- 繁忙时段的访问高峰需关注，确保系统稳定性

d. 用户统计：需要对注册用户数、活跃用户数等进一步细分，判断客户群结构变化及营销策略的效果。

- 新注册用户数及其后续转化率需要持续统计
- 活跃用户数及次要指标的变化需定期评估
- 不同年龄段及性别用户消费能力跟踪分析
- 流失用户的比例及可能的流失原因需要判断
- 老客户的回流情况及转化也需要重点跟踪

6. 权限管理：提供完备的后台权限管理机制，包括给管理员、编辑和其他角色进行权限控制。

a. 角色设定：需要事先设置好管理员、编辑、运营等不同角色。

- 管理员拥有平台最高权限，编辑次之，普通运营最低
- 不同角色的创建、变更及删除权限需要限制，确保安全
- 新增或调整角色时，需要评估其对现有权限产生的影响
- 角色的具体权限配置需要简洁明了，方便权限控制与判断

b. 模块权限：需要对订单、商品等不同模块进行权限控制。

- 不同模块对角色的授权粒度需详细配置：增删改查等操作
- 模块中涉及资金、资料等敏感数据的操作需严控权限
- 模块之间的相互影响也需要考虑，避免产生漏洞
- 模块改版升级时，需要再评估及测试其对权限的影响

c. 流程授权：需要对复杂流程进行整体的权限设计，如支付流程。

- 支付流程涉及结算周期设置、对账查核等操作，需要严密控制
- 退款流程涉及资金流转，也需要对审批环节进行权限划分
- 不同环节的数据变更会影响后续流程或汇总，这需要关注
- 流程调整优化后需要重新评估权限配置，确保安全

d. 操作审计：需要对敏感操作进行审计，跟踪操作人员与时间。

- 资金及资料的增删改操作需要进行审计跟踪
- 访问后台登录信息也需要进行审计，防止非法访问
- 审计日志需要定期核查，跟踪异常操作并作出响应
- 操作审计需保留相当长时期的数据，方便事后查核

e. 权限变更：角色或模块权限变更后，需要清理相关缓存，确保变更生效。

- 权限变更后需要清理权限认证相关缓存
- 可疑操作也需要在权限变更后再次核对，确保数据准确
- 权限变更记录同样需要保存，方便日后追溯及审计

7. 全程物流跟踪: 提供完备的物流跟踪过程, 包括物流公司选择、物流信息导入、客户物流查询等功能。

- a. 物流公司: 需要提供多家主流物流公司的选择, 满足不同需求。
 - 需要提供顺丰、申通、中通等多家公司的选择
 - 不同物流公司的运费及运输时效需配置明确, 提供客户选择依据
 - 新增物流公司需评估其服务质量与网络覆盖面, 确保客户体验
 - 需要与各大物流公司对接数据, 实时获取物流动态信息
- b. 发货创建: 商品发货时需要创建物流信息, 并关联具体订单。
 - 需要录入发货的物流公司、运单号、包裹数量等信息
 - 需要关联具体的订单信息, 确保物流信息的准确性
 - 需要支持多包裹发货, 并进行包裹级的信息填写
 - 物流信息的创建需要即时更新至各物流公司系统, 方便客户查询
- c. 物流跟踪: 需要提供物流轨迹的详细跟踪, 涵盖送达与签收。
 - 需要从发货开始实时展示物流的流转轨迹
 - 拆包、交接等事件也需要及时更新, 让客户知晓动态信息
 - 需要提供预计送达时间, 以供客户参考
 - 需要明确标识每次扫描的具体位置, 更直观展现物流流转
 - 签收后需要及时更新, 并提供签收图片或信息以确认签收
- d. 预警机制: 需要提供物流延误或异常的预警提示, 让客户知情。
 - 物流超过预计送达时间 24-48 小时未更新时, 需要预警客户
 - 物流发生异常事件如破损、丢失时, 需要第一时间通知客户
 - 提供人工客服或机器人进行二次确认, 解答客户问题
 - 根据事件严重程度, 需要采取相应补救措施, 确保客户权益
- e. 客户查询: 需要提供物流查询入口, 方便客户实时掌握物流信息。
 - 需要支持通过订单号、收货人姓名+手机号等多种条件进行查询
 - 查询结果需要直观显示当前物流状态及每一次扫描的流转信息
 - 查询次数较多时, 需要有次数限制或验证码校验, 避免恶意查询
 - 需提供人工客服或常见问题解答, 帮助客户及时理解物流信息

用户故事

作为一个购物者,我希望能够浏览产品目录并查看产品的详细信息和价格以选择并购买心仪的商品,将其添加到购物车并在需要时进行管理,以便在结账时一次性购买,我能够进行安全的在线支付以完成购买过程,查看订单历史记录和订单状态跟踪我的购买,我使用新人注册赠

券或分享赠券在平台购物,参与平台上的各种营销活动包括砍价、秒杀、拼团,选择和使用各种主流的支付渠道完成交易,查询自己所有订单的物流动态并进行跟踪以随时知晓位置。

作为一个商家,我管理产品目录包括添加、编辑和删除产品以及及时更新商品信息,管理订单包括查看和处理订单以满足客户需求,生成销售报告和统计数据分析业务绩效和趋势,跟踪商品的物流信息,上传商品图片和属性并实时更新商品信息,创建各类营销活动如砍价和拼团并跟踪活动效果,设置商品优惠券提高销量并分析各券种效果,查看商品的销量、评价和访问数据分析 and 优化策略。

作为一个管理员,我根据权限对商家、编辑和运营等不同角色进行授权,管理全流程结算包括结算周期设置、金额对账和复核,查看各类数据报表包括销售额、流量和用户等进行汇总分析,管理用户反馈的所有信息并迅速作出回应和改进。

作为商家运营成员,我需要分析各种销售数据以优化策略。

领域模型

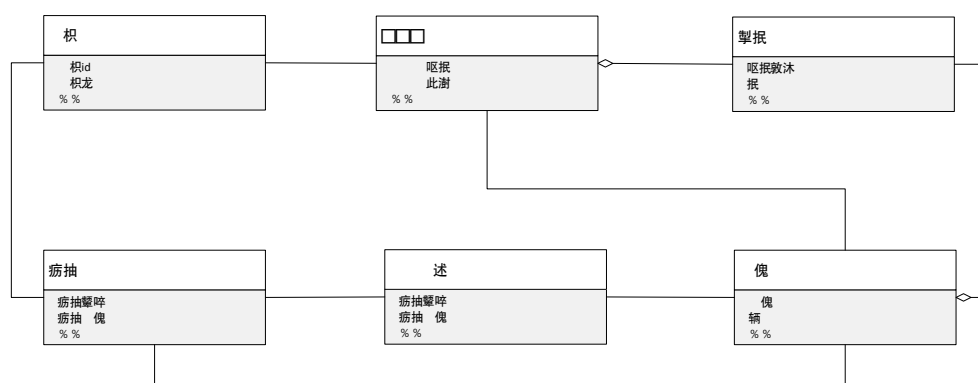
在电子商务平台的领域模型中,一般会涉及以下实体和关系:

1. 用户 (User): 购物者和管理员的账户信息。
 - 用户 ID: 用户在系统中的唯一标识
 - 用户名: 用户登录系统使用的用户名
 - 密码: 用户的登录密码
 - 注册时间: 用户注册系统的时间
 - 手机号: 用户的手机号码
 - 邮箱: 用户的电子邮箱地址
 - 审核状态: 用户账号的审核状态,如审核中、审核通过、审核不通过等
2. 产品 (Product):
 - 商品: 包含标题、图片、价格、库存、参数等属性
 - 商品分类: 以树形结构组织商品,方便客户浏览与过滤
 - 品牌: 作为商品的上级分类,代表特定品牌下的所有商品
 - 商品评论: 来自客户的商品评价,其他用户的参考信息

- 商品统计: 包含访问量、销量、评价等数据, 商家制定策略依据
- 3. 购物车 (Shopping Cart): 关联用户和产品, 记录用户选择的产品和数量。
 - 购物车: 存储客户选择的商品信息, 可以修改数量或删除
 - 购物车商品: 客户选择的具体商品, 包含商品信息、选择数量等
 - 购物车优惠: 平台提供的优惠券或活动, 在购物车中领取并使用
 - 购物车统计: 选中的商品总数、总价等信息, 方便客户购物决策
- 4. 订单 (Order): 关联用户和产品, 表示购买的产品信息和订单状态。
 - 订单流程: 从下单到支付、发货、收货等全过程
 - 支付: 使用微信支付、支付宝等渠道完成订单支付的过程
 - 发货: 商家完成订单中商品的打包与发出
 - 物流: 配送订单中的商品, 提供订单物流轨迹查询
 - 结算: 平台定期与商家对账结算金额, 主要依据订单交易数据
- 5. 支付 (Payment): 关联订单和支付信息, 记录支付方式和支付状态。
 - 支付方式: 平台提供的各支付渠道, 如微信、支付宝、货到付款等
 - 支付订单: 提交的订单信息及实付金额, 生成支付交易记录
 - 支付交易: 使用特定支付渠道支付的具体交易信息, 包括流水号等
 - 退款交易: 客户申请的退款交易, 同样需要生成新的交易记录
 - 支付统计: 各支付渠道交易额、交易量等统计数据, 用于销售额分析
 - 回调验签: 支付渠道回调需在平台服务器验证签名, 确保信息安全
 - 成功回调: 支付成功的回调, 需要更改订单状态并记录支付信息
 - 失败回调: 支付失败的回调, 需要标识订单支付失败, 同时通知用户
 - 异常回调: 支付渠道回调异常, 需要人工介入确认并作出相应处理
- 6. 营销模块 (Marketing): 这个模块提供创建与管理各类营销活动的功能, 目的是提高平台销量。
 - 优惠券: 提供部分商品现金优惠, 来吸引客户下单

- 会员：根据消费情况提供会员等级，享受会员专属优惠
- 拼团：提供部分商品的团购形式，通过群体购买达到优惠目的
- 秒杀：限时限量的商品促销手段，利用抢购心理提高销量
- 活动统计：各类活动参与人数、成交订单量等数据，评估活动效果

所画的图形如下



系统操作

1. 用户操作:

- 注册账户：输入用户名、密码、手机号、邮箱等信息完成注册
- 浏览产品目录：显示产品列表和相关信息，提供过滤与排序，用户可查看产品详情
- 查看产品详细信息：显示产品描述、价格、图片和其他属性，用户可加入购物车或返回目录
- 添加产品到购物车：在产品详情页点击“加入购物车”，将产品添加至购物车并显示数量
- 管理购物车：查看购物车内容，修改产品数量、移除产品或清空购物车
- 安全在线支付：提供信用卡、支付宝、PayPal 等支付方式，用户支付并确认订单
- 查看订单及状态：查看历史订单与当前订单，包括订单号、日期、状态与详情

2. 购物操作:

- 浏览商品：浏览商品分类和品牌下的所有商品信息

- 搜索商品: 输入关键词搜索想要的商品
- 添加商品到购物车: 选中想要的商品规格与数量, 添加到购物车
- 修改购物车: 在购物车中可以修改商品数量或移除商品
- 下单: 在购物车中勾选需要购买的商品, 进入填写订单详情页, 完成订单
- 支付订单: 在订单详情页选择支付方式, 完成订单支付

3. 管理员操作:

- 审核用户: 管理员需要审核用户提交的注册信息, 审核通过才可正常使用
- 面向商品管理: 添加、修改、删除商品信息, 设置商品价格与库存
- 商品分类管理: 添加、修改、删除商品分类, 设置分类层级关系
- 品牌管理: 添加、修改、删除品牌信息
- 订单管理: 查询所有订单, 修改订单状态, 发货与配送
- 活动管理: 创建、修改、删除各类营销活动, 设置活动使用规则
- 权限管理: 设置不同管理员的功能权限, 控制操作范围
- 数据统计: 统计会员总数、商品销量、订单完成率等数据, 便于商家制定经营策略
- 管理产品库存: 添加、修改和删除产品
- 管理订单: 管理员查看和处理订单, 更新订单状态, 生成销售报告

4. 支付操作:

- 支付订单: 在订单详情页选择微信支付、支付宝、货到付款等支付订单
- 退款申请: 用户可提出全额或部分退款申请, 平台审核通过后退还款项
- 异常处理: 出现支付回调异常, 需要人工干预审核订单支付状态, 确认实际支付结果

微服务拆分

微服务的拆分思路为: 根据业务能力进行服务拆分。

我认为对于商城的业务主要可以拆分为用户服务, 商品服务, 支付服务, 营销服务, 管理服务者几种微服务

1. 用户服务:

- 注册: POST /users
- 登录: POST /users/login
- 获取用户信息: GET /users/:id
- 修改用户信息: PUT /users/:id
- 添加购物车商品: POST /users/:id/cart

- 修改购物车商品: PUT /users/:id/cart
 - 删除购物车商品: DELETE /users/:id/cart
 - 下单: POST /users/:id/orders
 - 获取订单: GET /users/:id/orders
2. 商品服务:
- 获取商品分类: GET /products/categories
 - 获取品牌列表: GET /products/brands
 - 获取商品列表: GET /products
 - 获取商品详情: GET /products/:id 依赖: 用户服务下的获取用户信息 API
3. 支付服务:
- 获取支付渠道: GET /payments/channels
 - 订单支付: POST /payments/orders/:id/pay 依赖: 用户服务下的下单 API
 - 支付回调验签: POST /payments/callbacks/verify
 - 支付回调成功: POST /payments/callbacks/success
 - 支付回调失败: POST /payments/callbacks/fail
 - 申请退款: POST /payments/refunds
4. 营销服务:
- 发布优惠券: POST /marketings/coupons
 - 获取优惠券: GET /marketings/coupons
 - 优惠券领取: POST /marketings/coupons/:id/receive
 - 秒杀活动发布: POST /marketings/flash-sales
 - 获取秒杀活动: GET /marketings/flash-sales
 - 拼团活动发布: POST /marketings/group-buying
 - 获取拼团活动: GET /marketings/group-buying
5. 管理服务:
- 上架商品: POST /managements/products
 - 修改商品: PUT /managements/products/:id
 - 下架商品: DELETE /managements/products/:id
 - 获取订单: GET /managements/orders 依赖: 用户服务下的获取订单 API
 - 取消订单: PUT /managements/orders/:id/cancel
 - 发货: PUT /managements/orders/:id/ship
 - 完成订单: PUT /managements/orders/:id/complete
 - 添加管理员: POST /managements/admins
 - 获取管理员: GET /managements/admins

通信与数据模式设计

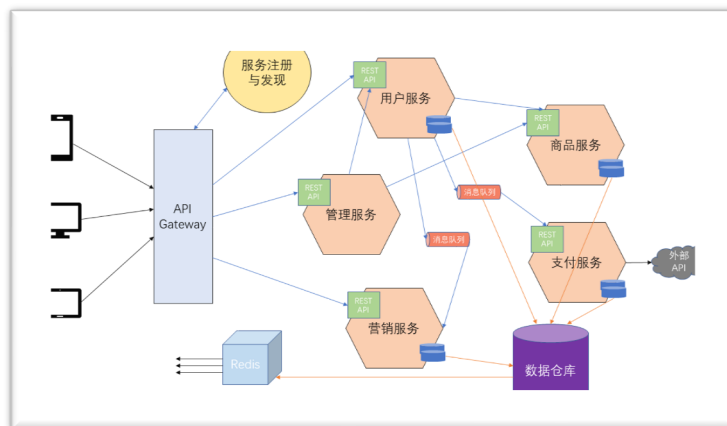
通信决策选择：

- **API Gateway:** 作为系统的统一入口，负责请求的转发、验证、限流等功能。可以针对不同客户端提供不同的 API，外部所有的请求都会先经过 API Gateway。
- **REST API:** 系统的部分非核心业务或日常管理业务使用的通信模式。
- **消息队列:** 主要用于用户服务、支付服务和营销服务之间的异步通信。例如，用户服务处理完下单请求后，可以通过消息队列通知营销服务去扣减优惠券数量，或者用户服务处理完下单请求后可以往消息队列发送一条订单创建的消息，支付服务监听到消息后去执行支付操作。
- **服务注册与发现:** 每个微服务启动时，会把自己注册到服务注册中心，微服务调用其他服务时，首先到注册中心查找服务地址，然后发起 HTTP 请求。

数据决策选择：

- 使用缓存 (Redis) 缓解数据库压力，加速查询，可以对访问频繁但不经常变更的数据进行缓存。
- 使用数据仓库整合不同微服务的数据，提供数据统一查询服务。
 - 建立全局数据模型，微服务定期推送自己的数据到数据仓库
 - 数据仓库定期拉取各微服务的数据，然后统一建模
 - 使用事件溯源的方式，记录每个微服务产生的业务事件，数据仓库从这些事件统一还原数据模型
- 使用 API 组合模式查询分散在多个微服务所拥有的数据库中的数据。

整体架构设计



微服务的部署和运维模式

(PPT 上描述的部署模式在此不再重复, 下面为部署或运维的完整可行方案)

1. 自动化部署: 通过持续集成和持续部署工具, 自动构建和部署微服务。
 - 持续集成: 微服务编码后, 通过 Jenkins 等工具自动构建和测试
 - 持续部署: 通过 Spinnaker、Argo CD 等工具自动部署微服务到 Kubernetes 等平台
2. 容器化部署: 将微服务封装在 Docker 或 Rkt 容器中, 然后在 Kubernetes 等容器管理平台部署和调度。
 - Dockerfile: 用于构建微服务 Docker 镜像
 - Kubernetes: 自动部署、扩展和管理容器化微服务
 - Helm: Kubernetes 的包管理器, 用于简化 Kubernetes 应用部署
3. 服务发现和注册: 通过 Eureka、Consul、Zookeeper 等服务注册中心实现微服务注册与发现。
 - 服务启动时向注册中心注册自身信息 (IP、端口、实例 ID 等)
 - 服务相互调用时, 先从注册中心获取对应服务的实例信息, 然后发起调用
4. 熔断和限流: 通过 Hystrix 等库对微服务进行熔断和流量控制。
 - 熔断: 某个微服务不可用时, 主动切断对其的调用, 避免级联故障
 - 限流: 控制对某个微服务的最大调用量, 避免其超负荷运行
5. 日志收集和监控: 通过 ELK 等日志收集工具和 Prometheus+Grafana 等监控工具监控微服务。
 - 日志收集: 收集各微服务的访问日志和错误日志
 - 监控: 监控微服务的调用情况 (请求量、成功率、耗时)、资源利用率 (CPU、内存)等
6. 零宕机部署: 通过蓝绿部署等手段实现微服务的零宕机发布。
 - 蓝绿部署: 新版本部署在一旁作为“绿色”环境, 等测试通过后再将流量切换过去, 以实现零宕机发布