

# 实习工作总结

## 新人项目

- 电商微服务项目
- 未使用内部中间件

## 活动

版本	负责人	奖励形式	风险点	预防方案
购物车买赠	finn.wang	买多免多，随单带走	<ul style="list-style-type: none"><li>• 免费品和付费品绑定关系出错</li></ul>	监控购物车操作报错打点
多单	keyes.qi、yopa.lu、habit.kuang	<ul style="list-style-type: none"><li>• 积攒星星，计算每单能带走的免费品</li><li>• 首单支付后100% rebate返券</li></ul>	<ul style="list-style-type: none"><li>• 星星和实际带走的免费品对不上</li><li>• 下单得的星星不对</li><li>• 首单支付后未发券</li><li>• 首件补仓逻辑导致roi过低，美国策略首件价值50刀以上</li><li>• 置顶品承接问题</li></ul>	
分天领统一发货	finn.wang、louis.chen	第一件免费品随单带走，剩余所有免费品团期结束后统一下零元单	<ul style="list-style-type: none"><li>• my gifts 页面异常，用户无法下单</li><li>• 用户最后没有带走所有已领取免费品</li></ul>	
二单硬拦截	yopa.lu、retro.you	将当前单免费品拆成两部分，一部分随单带走，一部分需要加到C端硬拦截门槛再支付一次带走	<ul style="list-style-type: none"><li>• 二单落地页异常，用户无法下单</li><li>• 二单下单后未带走所有免费品</li></ul>	
多单每单带走一件	yopa.lu	每单带走一件，每次要加到门槛才能带走当前免费品	<ul style="list-style-type: none"><li>• 多单落地页异常，用户无法下单</li><li>• 下单完后未带走所有免费品</li></ul>	
人质版	finn.wang	将当前单免费品拆成两部分，一部分随单带走，另一部分收到签收消息之后统一下零元单	<ul style="list-style-type: none"><li>• my gifts 页面异常，用户无法下单</li><li>• 未接收到签收消息，活动状态无法流转</li><li>• 用户最后没有带走所有免费品</li></ul>	
去除件数金额版本	retro.you、louis.chen	<ul style="list-style-type: none"><li>• 免费品随单带走</li><li>• 支付后发券，固定升级到100美元</li></ul>	<ul style="list-style-type: none"><li>• 支付后未发券</li></ul>	

## 多单得免费品

[【赠品】多单得免费品玩法 · Note](#)

## XY免单版本

[XY免单版本 · Note](#)

## 购物车买赠

[Note](#)

## 赠品

###

## 跳链

---

### 赠品

[https://www temu.com/free-items.html? bg\\_fs=1&ffcr=KilOsiAlwi& mkt\\_usr\\_p\\_from=push\\_all& i mmersive=77&rdct=1&is\\_back=1&skip\\_download=1](https://www temu.com/free-items.html? bg_fs=1&ffcr=KilOsiAlwi& mkt_usr_p_from=push_all& i mmersive=77&rdct=1&is_back=1&skip_download=1)

### 买赠

[https://www temu.com/bgnb\\_buy\\_cart\\_get\\_free.html? bg\\_fs=1&gi=XXR33Q7UMT7BOND3DHSDXQ XUHM&si=P5ESDFXVGEV2C7QIG4EHOEDAU& mkt\\_usr\\_p\\_from=push\\_all&skip\\_download=1](https://www temu.com/bgnb_buy_cart_get_free.html? bg_fs=1&gi=XXR33Q7UMT7BOND3DHSDXQ XUHM&si=P5ESDFXVGEV2C7QIG4EHOEDAU& mkt_usr_p_from=push_all&skip_download=1)

## 需求

---

### 1.【赠品】强化免费品页面文案

新增ab实验

### 2.【赠品】广告超价包告警逻辑调整 TVADS-799740

新增打点

### 3.5wk3【赠品】分天领支付后导流

ab实验调整，新增国家

导流逻辑....

### 4.【吉尔吉斯斯坦/斯里兰卡】开团动画优化：新增全场0元大字报加载动画

实验新增国家

### 5.【赠品】确认地址浮层样式改造

新增ab实验

[【赠品】确认地址浮层样式改造](#)

## 6.【赠品】强化大字报衔接

新增ab实验

[【赠品】强化大字报衔接 TVADS-529277 · Note](#)

## 7.【赠品】支付页高亮行动点

<https://brook.temu.team/requirement/TVADS-839912>

## 8.【赠品】商品加购底纹加件数

<https://brook.temu.team/requirement/TVADS-692763>

### 进度

线上验证

## 9.【赠品】新增渠道&新增渠道盐

[【赠品】新增渠道&新增渠道盐·Brook](#)

### 进度

没进测试

渠道盐实验

[授权弹窗下发实验详情](#)

[授权引导登录实验详情](#)

## 10.接入省钱助手渠道

[【赠品】接入新渠道省钱助手·Brook](#)

渠道盐实验

[实验详情](#)

示例

scene:com.temu.bg.jones.api.contract.enums.SceneEnum#SAVE\_ASSISTANT

jonesVersion: 0.1.35-save-SNAPSHOT

资格接口:

com.temu.bg.jones.api.contract.service.entrance.FreebiesEntranceService#queryEntranceQualify

sceneName =

uid

backendOrder =true

freeCount = -1

## 11.【赠品】砸金蛋换杯子三选一

[【日本】赠品砸金蛋换成杯子三选一·Brook](#)

### 进度

线上发布

## 12【赠品】欧区砸冰块开团

<https://brook temu.team/requirement/TVADS-419718>

预发测试

## 13【赠品】登录条异化文案导活动落地页

<https://brook temu.team/requirement/TVADS-858952>

不做!

## 14【赠品】PC/H5导下载页面优化

<https://brook temu.team/requirement/TVADS-226212>

不做!

## 15【澳大利亚】赠品-开团动画强调免费品价值

<https://brook temu.team/requirement/UN-464621>

## 16【赠品】系统风券奖励样式

<https://brook temu.team/requirement/TVADS-765704>

命中本次实验

1. /api/generic/jones/freebie/landpage/info + api/generic/jones/freebie/cart + api/generic/jones/freebie/cart/operate接口的cart\_info.pay\_status\_ab\_result新增 is\_show\_system\_pattern\_style: boolean

2. 支付页windowTag: PAY\_PAGE\_USER\_GUIDE, 新增  
data.ab\_result.is\_show\_system\_pattern\_style: boolean
3. REACHED\_THRESHOLD\_REBATE\_COUPON弹窗下发upgrade\_coupon\_boost\_amount字段(券升级差额)
4. MULTI\_FREEBIES\_FREE\_SHIPPING和PAY\_PAGE\_USER\_GUIDE两种, 命中本实验都返回  
PAY\_PAGE\_USER\_GUIDE

## 17.6wk3 【赠品金额版】checkout淡化券包利益点

<https://brook temu.team/requirement/UN-754948>

6wk4 【赠品金额版】checkout淡化券包利益点

<https://brook temu.team/requirement/UN-486220>

【多米尼加/危地马拉/萨厄瓦多】赠品金额版checkout淡化券包利益点

<https://brook temu.team/requirement/UN-894594>

## 18. 【赠品】统一预加载动画

<https://brook temu.team/requirement/TVADS-527919>

- 1.在当前已有加载动画的国家开
- 2.当命中本实验时, 且不是登陆后场景, 新增popup\_id, 在所有弹窗最前列; 登陆后场景不出
- 3.命中本实验室, 原有的预加载动画不生效
- 4.命中欧区合规实验时, 本实验不下发 (建议在原有预加载动画下层开本需求的实验)

加载动画实验

<https://bigdata temu.team/abtest/app/NWUSR/config/183883>

欧区合规实验

<https://bigdata-eu temu.team/abtest/app/NWUSR/config/208398?key=core>

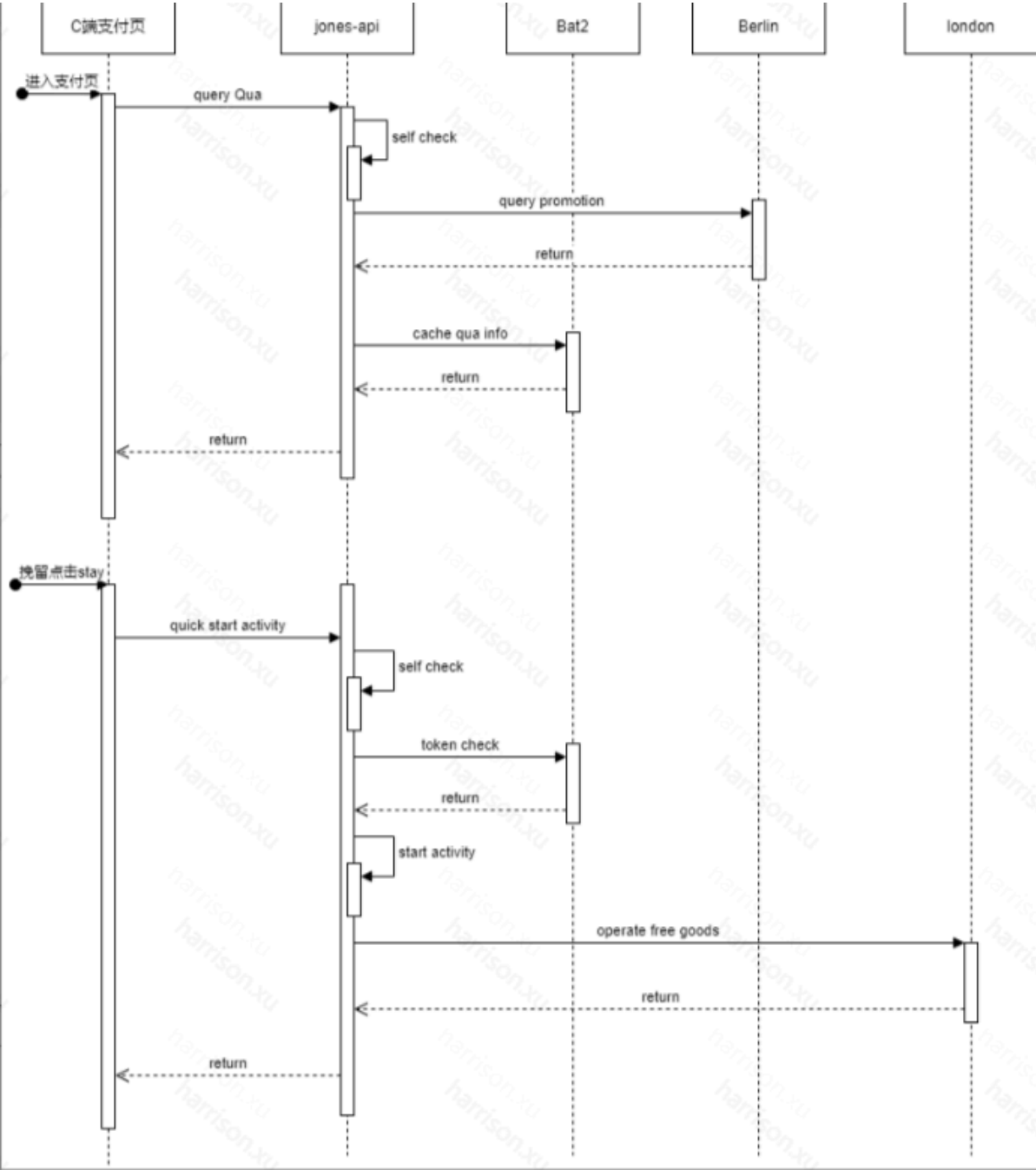
## 19.[B] TVADS-223734 【赠品】全球-商品加购底纹加件数

<https://brook temu.team/requirement/TVADS-223734>

To do: 除了澳大利亚外, 都推全 (韩国、日本、巴西、多米尼加、马来西亚、秘鲁、智利、厄瓜多尔、危地马拉、洪都拉斯、阿根廷、萨尔瓦多)

## 20. 【澳大利亚】C端下单页back出赠品利益点

<https://brook.temu.team/requirement/UN-783009>



实现

### 1.新增配置

TemuPayPageGiftConfig配置类

用于根据市场区域和用户类型（新用户或老用户）获取支付页面选择礼物的默认最小金额配置。它通过 regionDefaultMinAmountMap 存储区域相关的金额信息，并提供了一个方法 defaultMinAmount 来动态获取金额。

## 2.新增实体类

```
package com.temu.bg.jones.api.contract.dto.goods;
import java.io.Serializable;
import java.util.Map;
import lombok.Data;

@Data
public class TemuFrontSkuDTO implements Serializable {
    private Long goodsId;
    private Long skuId;
    private Integer goodsNumber;
    private Map<String, Object> extendMap;
}
```

## 3.新增场景

```
TEMU_PAY_PAGE_PICK("temu_pay_page_pick", "Temu Pay Page Pick",
SceneType.COMMON),
```

## 4.实体类

资格实体类

GiftEntranceResponse

QualifyTemuPayPageDetainResponse

## 5.GiftHandler类

```
package com.temu.bg.jones.api.service.gift.inner;
import com.temu.bg.jones.api.common.config.cache.StrategyConfigCache;
import com.temu.bg.jones.api.contract.enums.SceneEnum;
import com.temu.bg.jones.api.service.gift.bo.GiftEntranceQualificationContext;
import com.temu.bg.jones.api.service.qualification.bo.QualificationType;

public interface GiftHandler {
    SceneEnum getScene();
    QualificationType qualifyEntrance(GiftEntranceQualificationContext context);
    QualificationType qualifyLanding(Long uid);
    StrategyConfigCache.EventSn getEventSn(Long uid);
    String genUrl(Long uid);
}
```

两个实现TemuCartThresHoldHandler、TemuPayPagePickGiftHandler

方案一

### 方案二

## 优化方案

## 2.push流量预热接口

这次需求因为是正常流量，会走正常的group，基本没有流量，由于看repin的过盐接口trace都是加载耗时，怀疑因为接口冷启动导致加载时间长，所以把这次的过盐挪到push组里，避免了加载过程

## 方案二 trace Monitor

### 3.风控和过盐并行、减少耗时

## 21.【购物车买赠】推推荐品&数值支持算法调控

	+ com.temu.ba.corelli.common.exception.TubaSrcException	>	67	L oooooooooooooooooooooooooooooooooooooo g
	+ com.temu.arch.sumer.client.exception.SumerDeqradeException	>	46	L oooooooooooooooooooooooooo g
	+ com.temu.ba.corelli.common.stars.exception.QuenyCollectStarsGoodsListException	>	13	L oooooooooo g
	+ com.temu.ba.corelli.common.exception.TexasSrcException	>	6	L oo o o g
	+ java.lang.ClassNotFoundException	>	5	L o o o g
	+ com.temu.ba.corelli.common.exception.GinSrcException	>	2	L
28	+ com.temu.ba.corelli.integration.impl.TubaIntegrationImpl\$TubaRecallException	>	61990	L oooooooooooooooooooooooooooooooooooooo g

## 22 xy挽留弹窗增加授权checkbox资源位

### 23.【购物车买赠】遗留-尝试硬门槛拦截

[B] TVADS-413342

## 24. 【澳大利亚】赠品接入购物车买赠版本

<https://brook.temu.team/requirement/UN-531534>

## MarketPriceDisplayByScene



## 25.遗留 - 【赠品】后置导登录

[\[B\] TVADS-511778](#)

## 26. 【购物车买赠】接低库存标&小数点后两位缩小

[\[B\] TVADS-478001](#)

上下游确认流量大小

### 1、商品价格样式

主接口ab\_result下发实验字段price\_style

price\_style=v1, 价格缩小后两位

price\_style=v2, 价格缩小后两位&划线价在前

### 2、低库存标

主接口ab\_result下发实验字段is\_show\_low\_inventory\_tag\_in\_goods\_list、  
is\_show\_low\_inventory\_tag\_in\_cart

商品的tags\_info下发goods\_tags (goods\_tags中的字段参考c端)

### 3、local标

主接口ab\_result下发实验字段is\_show\_local\_tag: true / false

## 27.赠品】加拿大随单领改分天领的弹窗位置调整

<https://brook temu.team/requirement/TVADS-204225>

## 28. 【赠品】广告渠道开团噱头包装

<https://brook temu.team/requirement/TVADS-656285>

## 29.7. 【葡萄牙】 赠品导登录样式优化

<https://brook temu.team/requirement/UN-519368>

## 30.[B] TVADS-493232 【购物车买赠】 算法需求 - 支持更多件数

31. [【新人免费领】购物车场景接入工作台·Brook](#)

32. **【赠品分单领高ROI】 100%back无上限，加兜底**

<https://brook temu.team/requirement/TVADS-343193>

33. [\[B\] TVADS-351884](#) **【购物车买赠】接入首页弹窗新渠道 - 设计中**

34. **【赠品】主版本新增客服资源位，判断活动状态，推送活动卡片**

<https://brook temu.team/requirement/TVADS-834634>

35. [\[B\] TVADS-653668](#) **【赠品】 100% benifit / back改100%bonus**

36. **【沙特阿联酋以色列+澳大利亚+乌克兰】赠品加满后直接跳转支付页**

<https://brook temu.team/requirement/UN-739134>

37. (两周) **【哥伦比亚+厄瓜多尔+巴拿马+哥斯达黎加+阿根廷】赠品首页弹窗免责文案修改**

<https://brook temu.team/requirement/UN-496470>

**【赠品】部分免费品确认收货后发放**

<https://brook temu.team/requirement/TVADS-464004>

[brook·Brook](#)

**【赠品】广告指定利益点切换为原价逻辑**

[【赠品】广告指定利益点切换为原价逻辑·Brook](#)

[brook·Brook](#)

## 土耳其分单支付

[【后台】土耳其引导分单支付·Brook](#)

[【赠品后台】土耳其引导分单支付·Note](#)

## 【赠品】统一发货新逻辑 逻辑二期国家兼容

## 消息渠道新起二级分发

[TVADS-632738](#) 【赠品多单】消息渠道新起二级分发(开发中)

[B] [TVADS-791506](#) 【购物车买赠】先选免费-EEA版

## 【赠品】免费商品领取增加硬拦截门槛

<https://brook.temu.team/requirement/TVADS-262461>

## 【赠品多单】下单后增加补仓逻辑

<https://brook.temu.team/requirement/TVADS-522095>

## 【赠品】延时支付创单后直接导秒杀

## 【活动】赠品&砍价导流秒杀，实验组用户回传标记给客服工作台

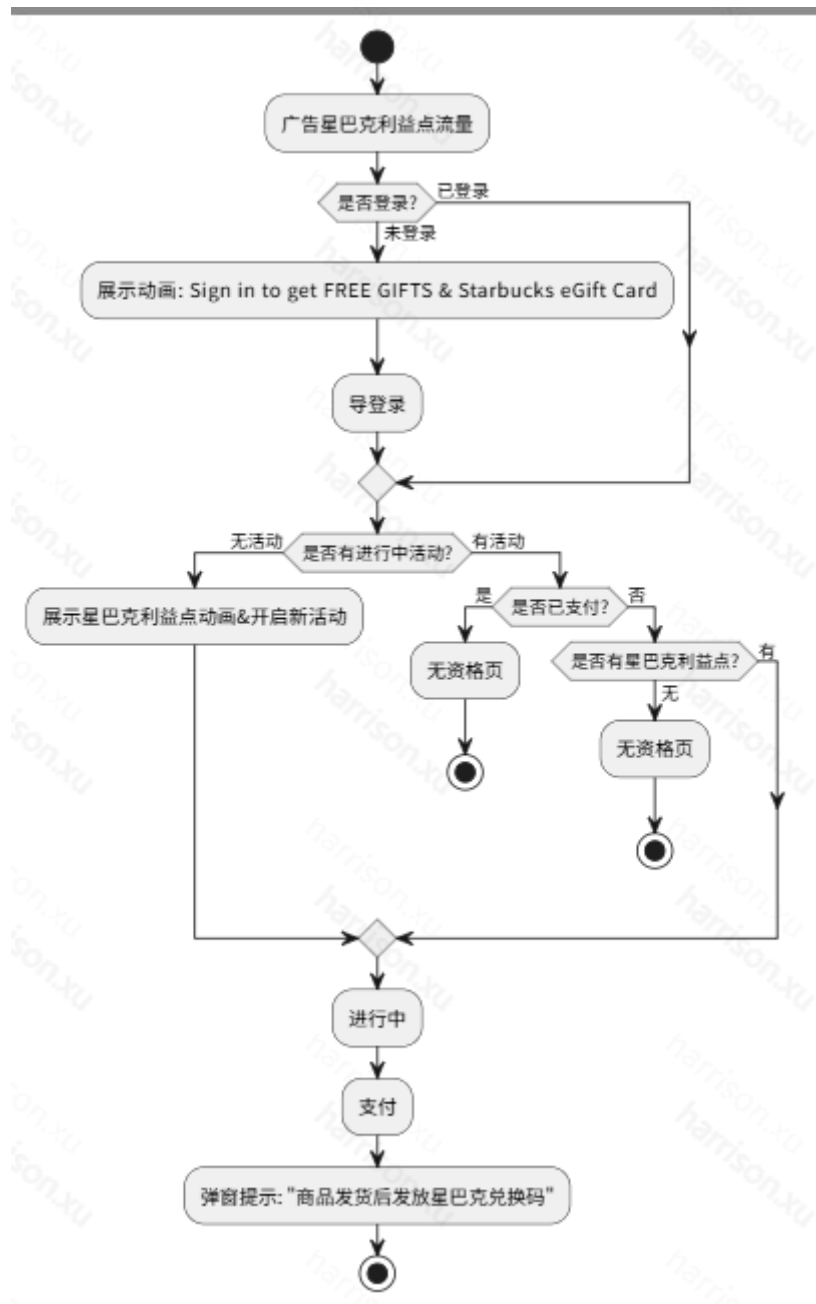
## 首件商品补仓动画优化

<https://brook.temu.team/requirement/686b405cd373cc2ac5552f95>

## 韩国星巴克

项目【temu-service/bg-jones】需求【【韩国-广告&新人】dn5赠品活动支持星巴克卡券利益点】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1655855/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件，目前仅支持Jetbrains系列IDE)，即可进行Code Review。

执行流程图



奖励插入，发奖，对账

7 月 CR

项目【temu-service/bg-jones】需求【【赠品前台】切x\_reduce\_y\_cash\_front\_goods】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1619930/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件，目前仅支持Jetbrains系列IDE)，即可进行Code Review。

项目【temu-service/bg-jones】需求【【日韩】赠品落地页接入品牌标】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1606229/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件，目前仅支持Jetbrains系列IDE)，即可进行Code Review。

@joie.lin @keyes.qi @retro.you @yopa.lu @finn.wang 预发测试完成，佬们帮CR下，发线上

项目【temu-service/bg-corelli】需求【【赠品多单】付费品阶段叠加100% back玩法】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1613466/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @keyes.qi @yopa.lu 应该没什么改动了 老板们帮看下

项目【temu-service/bg-jones】需求【【土耳其】赠品导登录优化】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1618647/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @keyes.qi @retro.you @yopa.lu @finn.wang 修改sumer文案拼接的逻辑, 大佬们帮忙review下, 准备上线

项目【temu-service/bg-corelli】需求【【活动】收集星星得免费品-接入工作台】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1613967/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @keyes.qi @yopa.lu 佬们求CR

项目【temu-service/bg-corelli】需求【【技术优化】多单新增校验和监控】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1618256/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@yopa.lu @joie.lin @habit.kuang  
大佬们帮忙cr下

项目【temu-service/bg-corelli】需求【【starback】数值策略调整】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1610820/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @yopa.lu @habit.kuang 求cr, 翻译给了, 替换下

项目【temu-service/bg-jones】需求【【赠品】统一发货新逻辑逻辑二期国家兼容】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1605845/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @louis.chen mygift开国语言校验改造, 大佬审批下

项目【temu-service/bg-corelli】需求【【赠品多单】增加升级动画承接制定件数利益点】Code Review 链接 <https://brook.temu.team/waterdrop/legacy-page/generalCodeReview/branchType/feature/1608259/diffs> 或复制本条消息后切换到IDE(需安装最新版本Cowizard插件, 目前仅支持Jetbrains系列IDE), 即可进行Code Review。

@joie.lin @yopa.lu @habit.kuang qcr

涉及

ab实验配置、新增ab实验、leo配置修改、日志系统排错、客户端冒烟测试

线上发布流程

赠品开国sop

## 问题记录

### 1 约旦开国配置

6.12 线上配置 npe

开国sop字段顺序和leo不一样

### 2 Spring AOP 循环引用详细解释

您遇到的循环引用错误是 Spring AOP 代理对象序列化时的一个典型问题。让我详细解释这个循环依赖是如何产生的：

#### 1. Spring AOP 代理结构

Spring AOP 使用代理模式来实现面向切面编程。当您的 `GinIntegrationImpl` 类被标记为 `@Component` 并被自动注入到其他类中时，Spring 实际上注入的是一个代理对象，而不是原始类的实例。

代理对象的命名遵循格式：`原始类名$$EnhancerBySpringCGLIB$$随机字符串`，比如错误信息中的 `GinIntegrationImpl$$EnhancerBySpringCGLIB$$e7aa1781`。

#### 2. 代理对象的内部结构

这个代理对象内部包含了复杂的结构，其中最重要的是：

- **advisors**: 管理所有应用于这个对象的 AOP 切面
- **targetSource**: 指向被代理的原始对象
- **proxiedInterfaces**: 代理实现的接口列表

#### 3. 循环引用的形成路径

从错误信息来看，循环引用的具体路径是：

```
GinIntegrationImpl$$EnhancerBySpringCGLIB$$e7aa1781["advisors"]
-> InstantiationExceptionAwarePointcutAdvisorImpl[1]
-> InstantiationExceptionAwarePointcutAdvisorImpl["declared_pointcut"]
-> AspectJExpressionPointcut["class_filter"]
-> 回到自身
```

这表明：

1. AOP 代理对象有一个 `advisors` 数组
2. 数组中的第二个元素 (索引 1) 是 `InstantiationExceptionAwarePointcutAdvisorImpl` 对象

3. 这个对象有一个 `declared_pointcut` 字段，指向 `AspectJExpressionPointcut`
4. `AspectJExpressionPointcut` 有一个 `class_filter` 字段，它又引用回到原始对象

这形成了一个引用环，Jackson 在序列化时不断遍历这个环而无法完成。

## 4. 为什么会出现这个问题？

当 Spring 应用 AOP 切面到您的 bean 时，它创建一个代理对象并配置它的 `advisors` 来包含所有应用的切面。问题在于这些切面的内部实现中包含了类型过滤器、表达式等，它们需要引用回原始类型或其他相关组件。

具体来说，`AspectJExpressionPointcut` 的 `class_filter` 需要能够确定哪些类是应该被该切面处理的。为了做到这一点，它需要持有对目标类或类加载器的引用，这就形成了循环。

## 5. Jackson 序列化中的问题

Jackson 默认会尝试序列化对象的所有字段。当它遇到 Spring AOP 代理对象时，它会：

1. 尝试序列化 `advisors` 数组
2. 序列化数组中的每个元素
3. 对于 `InstantiationModelAwarePointcutAdvisorImpl`，它尝试序列化 `declared_pointcut`
4. 继续序列化 `AspectJExpressionPointcut` 的 `class_filter`
5. `class_filter` 又引用回原始对象，形成循环

Jackson 检测到这个循环后抛出 `InvalidDefinitionException` 异常。

## 6. 为什么这是 AOP 的问题而不是业务代码的问题

Spring AOP 的内部结构天然包含这种循环引用，这不是您的业务代码造成的。当您的日志系统尝试序列化整个代理对象时，就会遇到这个问题。

Spring 框架本身不需要序列化这些代理对象，所以这种循环引用在正常使用中不会造成问题。只有在尝试将代理对象转换为 JSON 或其他序列化格式时才会出现问题。

## 7. 常见解决方案

对于这种 Spring AOP 代理序列化问题，有几种解决方案：

1. **避免序列化代理对象**：最简单的方法是在日志记录时识别并特殊处理代理对象，只记录基本信息。
2. **配置 Jackson 忽略问题字段**：通过 `@JsonIgnore` 或 Jackson Mixins 配置忽略 `advisors`、`targetSource` 等字段。
3. **修改序列化策略**：配置 `ObjectMapper` 允许自引用或修改循环引用的处理方式。
4. **获取原始目标对象**：如果可能，通过 `AopUtils.getTargetClass()` 或 `((Advised)proxy).getTargetSource().getTarget()` 获取被代理的原始对象进行序列化。

在您的情况下，最合适的解决方案是修改日志记录逻辑，避免尝试序列化 Spring AOP 代理对象的完整内容，而只记录其基本信息，正如我在之前的回答中建议的那样。

3.Bat2问题记录

[历史事故 · Note](#)

# 支付页主接口优化

根据你提供的 diff，这次提交的核心目标是**赠品支付页主接口性能优化**。下面是详细分析：

## 1. 优化点核心思路

### 主要优化方式

- 引入 SPI 透传数据 (spiExtData) 机制，在支付页主接口的调用链中，优先复用上游已计算好的购物车上下文和数据，避免重复的业务计算和数据库查询。
- 通过 Map 结构 (spiExtMap) 在接口间传递序列化后的业务对象 (如 CartContext、CartInfoDTO)，下游直接复用，提升接口整体性能。

## 2. 关键代码变更说明

### (1) 主接口逻辑优化

以 `TemuBackendOrderServiceImpl` 为例：

```
// 优化前
Pair<CartInfoDTO, CartContext> cartContextPair =
    freebiesCartLogic.queryCartAndContext(uid, activityDetail);

// 优化后
FreebiesObamaSpiExtDataBO spiExtDataBO = (FreebiesObamaSpiExtDataBO)
    MapUtils.getObject(request.getSpiExtMap(),
        CommonConstants.FREEBIES_SPI_EXT_DATA);
Pair<CartInfoDTO, CartContext> cartContextPair =
    Optional.ofNullable(spiExtDataBO)
        .map(data -> data.getCartContextPair())
        .orElseGet(() -> freebiesCartLogic.queryCartAndContext(uid, activityDetail));
```

解释：

- 优先从 request 的 `spiExtMap` 里获取上游透传的购物车数据（如果有），没有再走原有的逻辑重新查询。
- 这样可以减少一次数据库/缓存/业务计算，提升接口响应速度。

### (2) spiExtMap 透传机制

- 在各业务流程（如日常赠品、N次赠品、多赠品等）中，构建和传递 `spiExtMap`，并将业务对象（如 `CartContext`、`DailyGiftCartContext`）序列化后放入 map。
- 新增了多个 BO 类（如 `FreebiesObamaSpiExtDataBO`、`DailyGiftFirstObamaSpiExtDataBO`、`DailyGiftNOrderObamaSpiExtDataBO`）用于封装透传数据。



### (3) 通用常量和埋点

- 在 `CommonConstants` 中新增了相关 key，方便统一管理和使用。
- 在 `CatEnum` 中新增埋点类型，便于监控优化效果。

### (4) 工具方法适配

- `OrderSpiCommonUtils`、`FreebiesGoodsUtils` 等工具方法适配了新的数据结构和透传机制。

---

## 3. 性能优化效果

- **减少重复计算**：主接口优先复用上游已算好的购物车上下文，避免重复的业务逻辑和数据查询。
- **提升接口响应速度**：减少了数据库/缓存/服务调用次数，接口整体性能提升。
- **埋点监控**：通过 `CatEnum` 埋点，实时监控优化分支的命中情况和效果。

---

## 4. 建议

本次赠品支付页主接口性能优化的核心思路是引入 SPI 透传机制，将上游已计算好的购物车上下文和数据通过 `map` 结构在接口间透传，下游接口优先复用，避免重复的业务计算和数据库查询。这样大幅提升了接口的整体性能和响应速度。同时通过埋点监控优化效果，确保优化分支的命中和正确性。

# Integration

根据你提供的 `integration` 文件夹结构，这是一个典型的**外部系统集成模块**，负责与各种第三方服务或内部微服务进行交互。下面是对这个模块的详细分析：

---

## 1. 模块整体架构

### 核心组件

- `IntegrationContext.java` - 集成上下文管理，可能是整个集成模块的核心配置和上下文管理类
- `AlgoIntegration.java` - 算法集成，可能是最大的集成类，负责算法相关的服务调用
- `RepinIntegration.java` - 推荐系统集成
- `GinIntegration.java` - 商品服务集成

### 子目录结构

- `impl/` - 具体实现类
- `ab/` - AB测试相关集成
- `bo/` - 业务对象
- `helper/` - 辅助工具类
- `rpc/` - RPC调用相关
- `producer/` - 消息生产者

- `utils/` - 工具类
  - `risk/` - 风控相关集成
  - `edox/` - 特定业务集成
  - `config/` - 配置相关
  - `aop/` - 面向切面编程
- 

## 2. 集成服务分类

---

### 业务服务集成

- 商品相关: `GinIntegration` (商品服务)
- 推荐相关: `RepinIntegration` (推荐系统)
- 算法相关: `AlgoIntegration` (算法服务)
- 风控相关: `MorganIntegration`, `RiskIntegration`
- 订单相关: `ObamaIntegration` (订单服务)

### 基础设施服务

- AB测试: `ab/` 目录下的AB测试集成
- 消息队列: `producer/` 目录下的消息生产者
- 配置中心: `config/` 目录下的配置管理

### 第三方服务

- 支付相关: `ThanosIntegration`, `ValenciaIntegration`
  - 物流相关: `FreightIntegration`
  - 用户相关: `SonaIntegration`
  - 营销相关: `SupremeIntegration`
- 

## 3. 设计模式分析

---

### 统一接口模式

每个集成类都遵循相似的命名规范 (`xxxIntegration`)，可能实现了统一的接口或抽象类，便于统一管理和扩展。

### 上下文管理

`IntegrationContext` 作为核心上下文类，可能负责：

- 服务发现和路由
- 连接池管理
- 配置管理
- 异常处理

## 分层架构

- **接口层:** 各个 `xxxIntegration` 类
- **实现层:** `impl/` 目录下的具体实现
- **工具层:** `utils/`, `helper/` 目录下的辅助工具
- **配置层:** `config/` 目录下的配置管理

"我们项目的 integration 模块采用了统一的分层架构设计，将所有外部系统集成统一管理。每个集成类都遵循 `xxxIntegration` 的命名规范，通过 `IntegrationContext` 进行统一的上下文管理。模块按功能分为业务服务集成（如商品、推荐、算法）、基础设施服务（如AB测试、消息队列）和第三方服务（如支付、物流）三大类。这种设计的好处是便于统一管理、易于扩展和维护，同时通过分层架构保证了代码的清晰性和可测试性。"

## 4. 技术特点

- **微服务架构:** 与多个内部微服务进行集成
- **第三方服务:** 集成各种外部服务
- **统一管理:** 通过统一的接口和上下文管理
- **可扩展性:** 模块化设计，便于添加新的集成服务
- **配置化:** 通过配置管理不同环境的服务地址和参数

## Tiger

项目中大量使用 Tiger 作为内部消息队列，实现业务的异步解耦。以 task 模块为例，所有需要异步处理的任务（如活动结束、订单变更、补偿等）都会通过 Tiger 发送消息。每个业务场景都有独立的消费者类，这些类继承自 `TigerBaseConsumer`，统一封装了消费者的初始化、监听和重试机制。具体业务只需实现 `process` 方法即可，极大提升了开发效率和系统健壮性。

Tiger 消费者支持自动重试、延迟消费、日志追踪等特性，保证了消息处理的可靠性和可观测性。通过这种方式，我们实现了业务流程的高效解耦和异步扩展。

这是面试中关于消息队列（如 Tiger/Kafka/RocketMQ 等）非常常见的高频问题。结合你项目中 Tiger 的实现，可以这样回答：

## 1. 如何保证消费不重复（幂等性）

- **业务层面实现幂等性：**

虽然 Tiger/Kafka 等消息队列本身只能保证“至少一次”投递，不能绝对避免重复消费，但我们可以在业务处理时保证幂等性。例如：

- 每条消息有唯一业务ID（如订单号、事件ID），消费前先查数据库/缓存是否已处理过，已处理则跳过。
- 关键操作（如扣减库存、发放奖励）采用唯一约束或分布式锁，防止重复执行。

- **消费端日志与状态记录：**

消费成功后记录消费状态，失败时重试，确保每条消息最终只被成功处理一次。

## 2. 如何保证可靠性（不丢消息）

- 消息队列本身的可靠机制：**  
Tiger/Kafka 支持消息持久化、ACK机制，只有消费端返回成功（如 `CONSUME_SUCCESS`），消息才会被确认消费。
- 消费端异常处理与重试：**  
在 `TigerBaseConsumer` 的 `listener` 里，捕获异常时返回 `RECONSUME_LATER`，Tiger 会自动重试，直到消费成功或达到最大重试次数。
- 消费失败告警与补偿：**  
达到最大重试后可触发告警，人工或自动补偿，确保消息最终被处理。

## 3. 如何保证顺序性

- 分区有序：**  
Tiger/Kafka 支持分区顺序消费。对于需要顺序的业务（如同一订单的状态流转），可以将同一业务ID的消息发送到同一个分区，消费端单线程消费该分区，保证顺序。
- 消费端单线程处理：**  
对于强顺序要求的 topic，消费端可以配置为单线程消费，或业务上串行处理同一 key 的消息。
- 业务幂等+顺序校验：**  
即使偶有乱序，也可通过业务幂等和状态机校验，防止状态回退或错乱。

## 4. 表达建议

“在我们项目中，针对 Tiger 消息队列的消费，我们主要通过业务幂等性保证不重复消费，通过队列的 ACK 和自动重试机制保证可靠性。对于有顺序要求的业务，我们会将同一业务ID的消息路由到同一分区，并在消费端串行处理，确保顺序性。即使出现极端情况，也会通过幂等校验和补偿机制兜底，保证业务数据的最终一致和正确。”

# Bat

## 主要接口解读

主接口 `getLandPage`

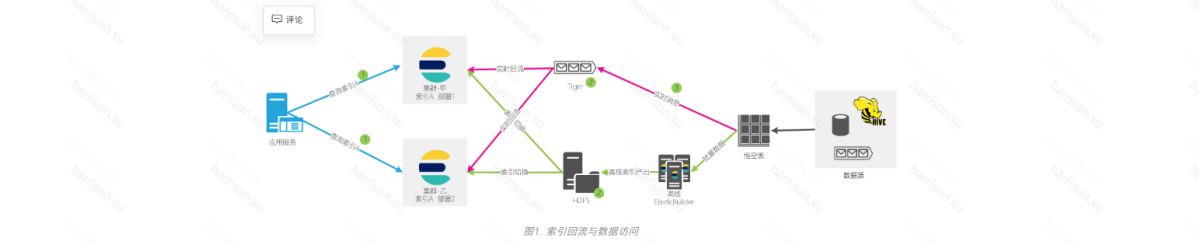
us-default/freebies/landingService.getLandPage	秒级分析	0%	0%	9,838,967	130.87	0	0	7.45	2,391.28	169.66	310.11	643.31	0.583
--	------	----	----	-----------	--------	---	---	------	----------	--------	--------	--------	-------

1.PES索引回流

PES索引回流是基于“离线索引构建技术”的Elasticsearch数据导入、数据访问服务。它是“Linda”、“ies快速导入”的迭代升级。

与以往数据导入平台相比，PES索引回流的显著特点就是批次（全量、批次增量）导入是在离线做的，不影响在线集群。因此变更shard数、修改mapping、增加源表等以往需要re-index的操作，现在都很简单方便，只需要在页面点击，重新触发一次回流。

典型场景：“我有几份异构数据在那里。我想用pes的查询方式便利地使用它”。数据源是必不可少的。



```
newbieEventSendHelper.sendGetLandPage(context);
```

operateCart

us-default/FreebiesLandingService.operateCart	秒级分析	-2.77%	17.99%↑	7,220,162	96.04	69	0.001	5.01	1,487.56	139.92	260.41	404.21	0.428
---	------	--------	---------	-----------	-------	----	-------	------	----------	--------	--------	--------	-------

queryCart

+ us-default/FreebiesLandingService.queryCart	秒级分析	-1%	-99.99%	5,048,238	67.15	5	0.0001	3.57	2,132.63	113.93	240.21	358.11	0.299
---	------	-----	---------	-----------	-------	---	--------	------	----------	--------	--------	--------	-------

queryNoticeQualify

GLO

+ notice/FreebiesNoticeEntranceService.queryNoticeQualify	秒级分析	3.14%	28%↑	424,374,917	5,644.56	4,031	S F	0.001	4.46	2,467.89	20.02	33.31	88.11	25.141
---	------	-------	------	-------------	----------	-------	-----	-------	------	----------	-------	-------	-------	--------

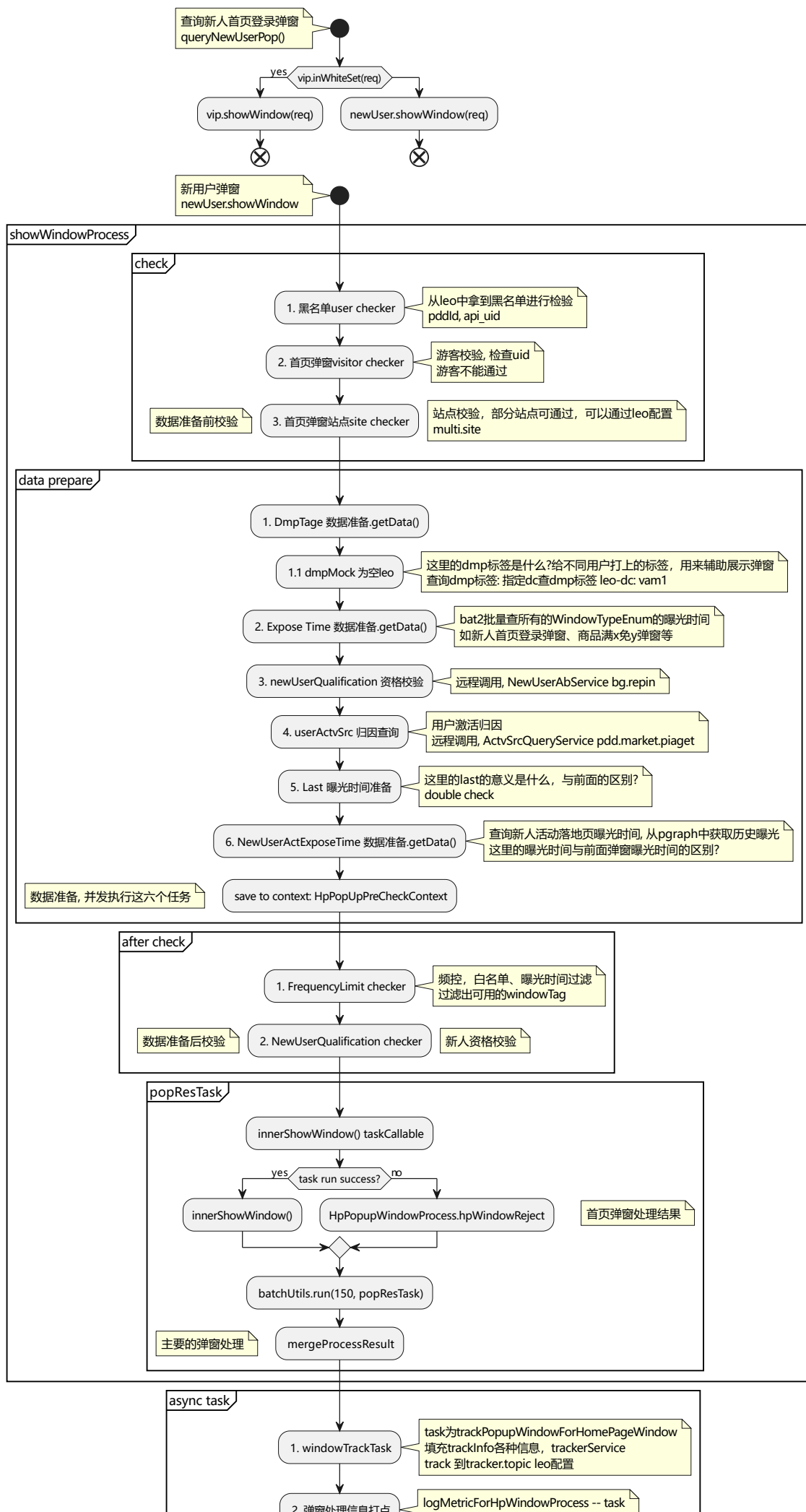
EU

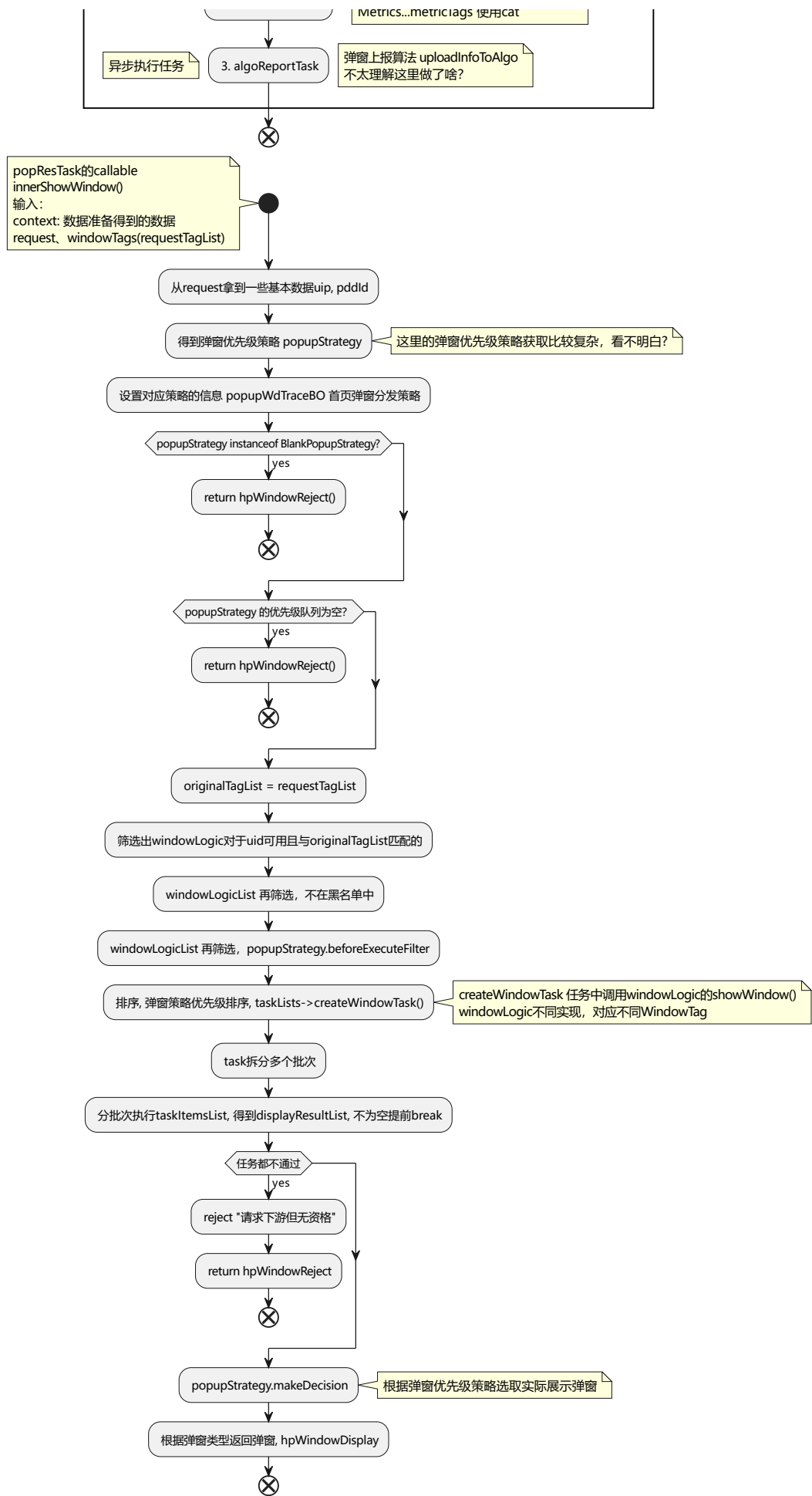
+ notice/FreebiesNoticeEntranceService.queryNoticeQualify	秒级分析	1.32%	3354.99%↑	628,025,382	8,336.66	1,711	S F	0.0003	3.5	1,400.42	26.46	38.5	140.31	28.537
---	------	-------	-----------	-------------	----------	-------	-----	--------	-----	----------	-------	------	--------	--------

qualifyEntranceQualify

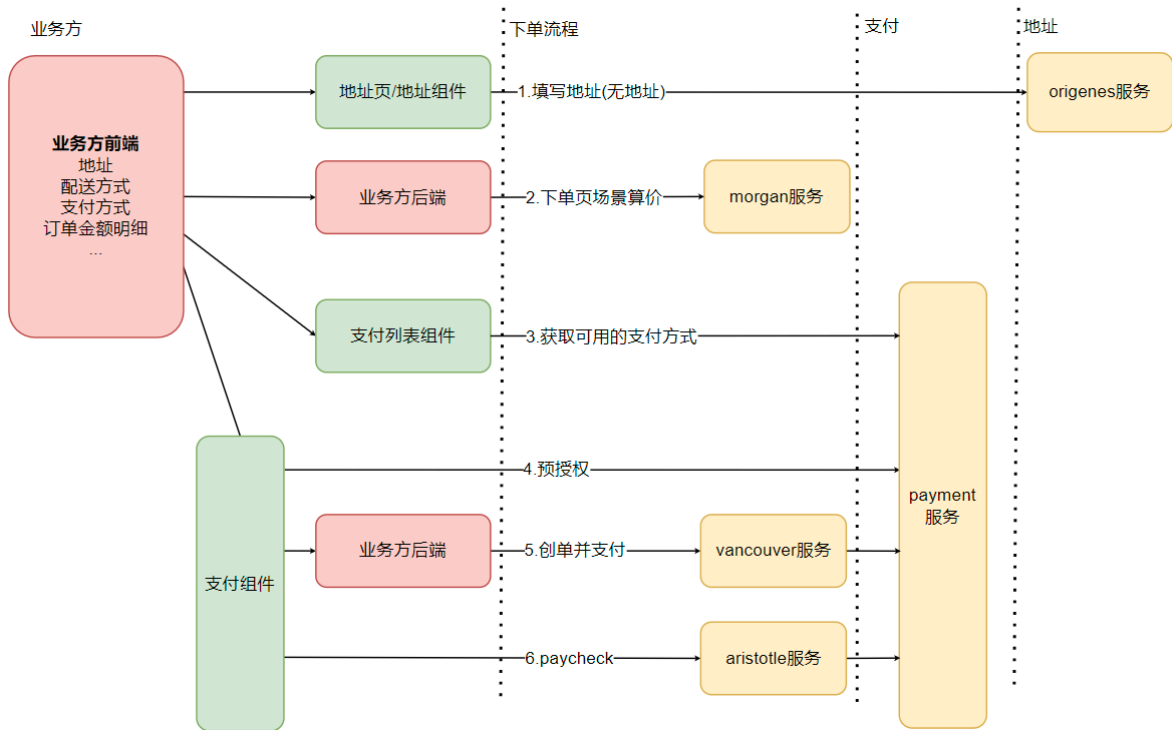
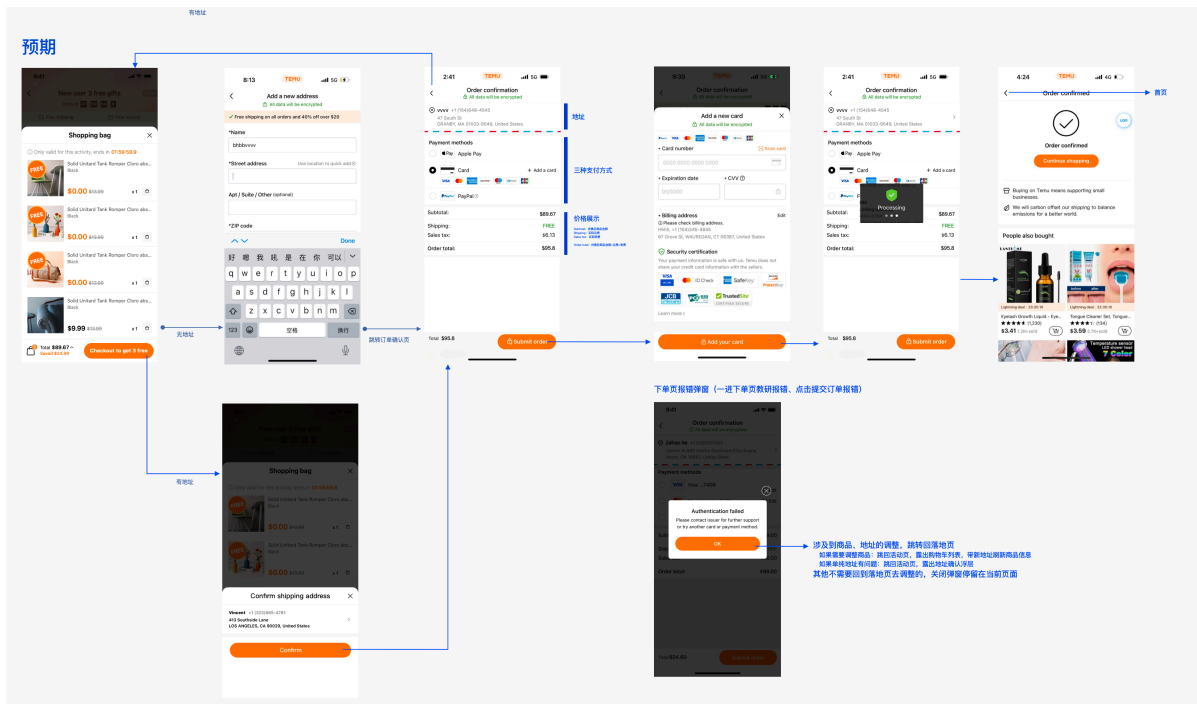
支付接口

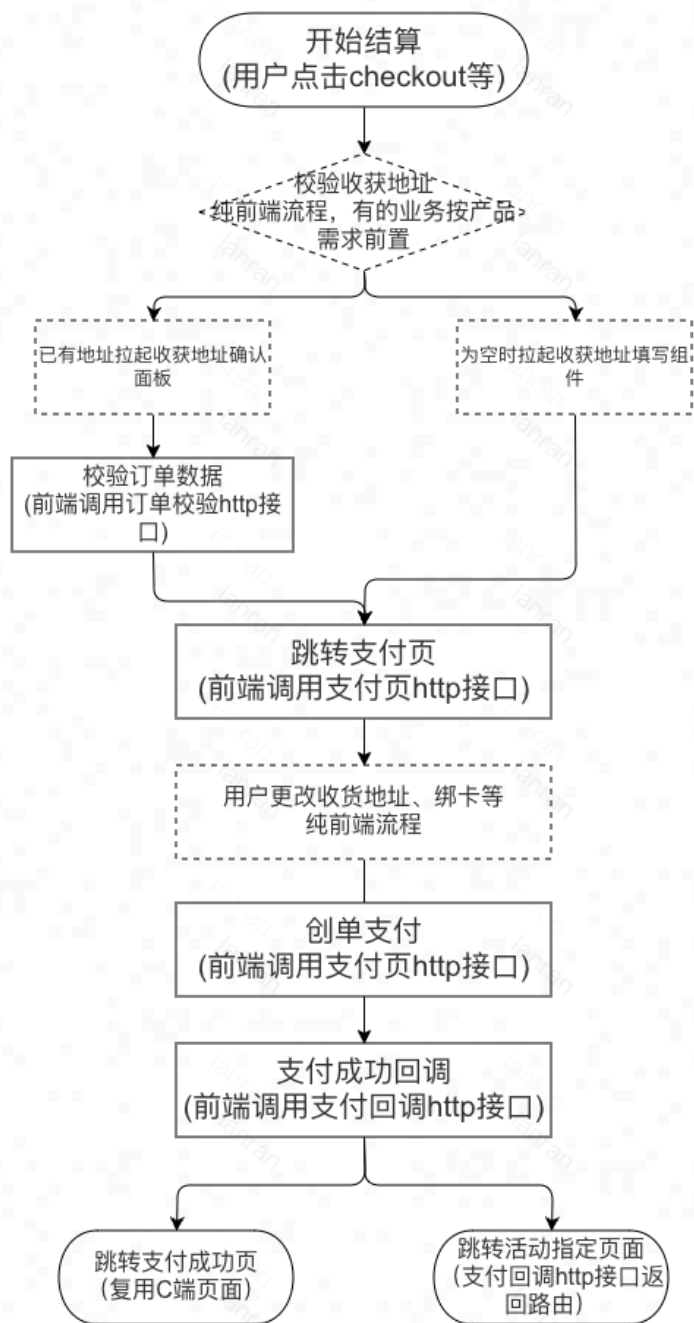


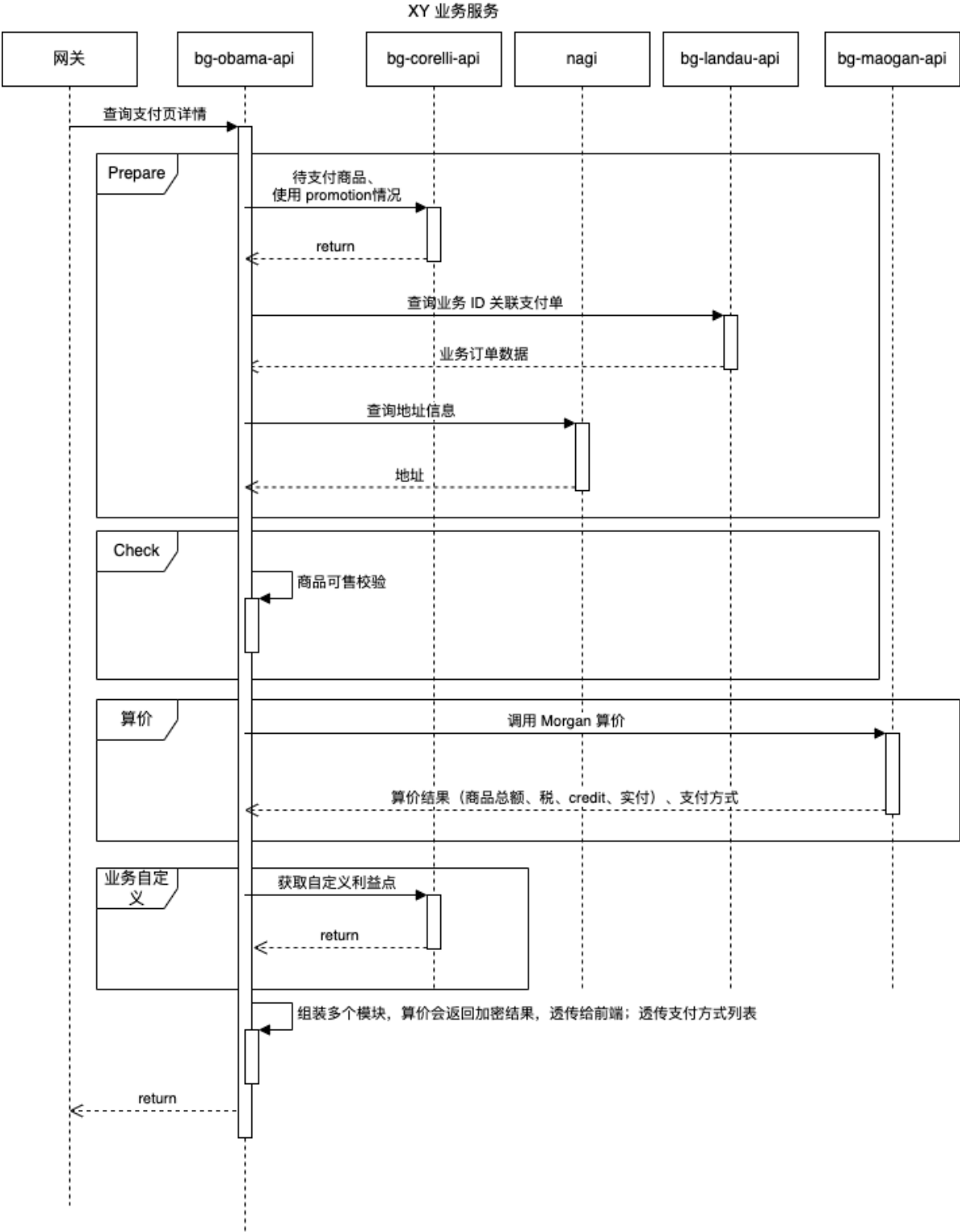


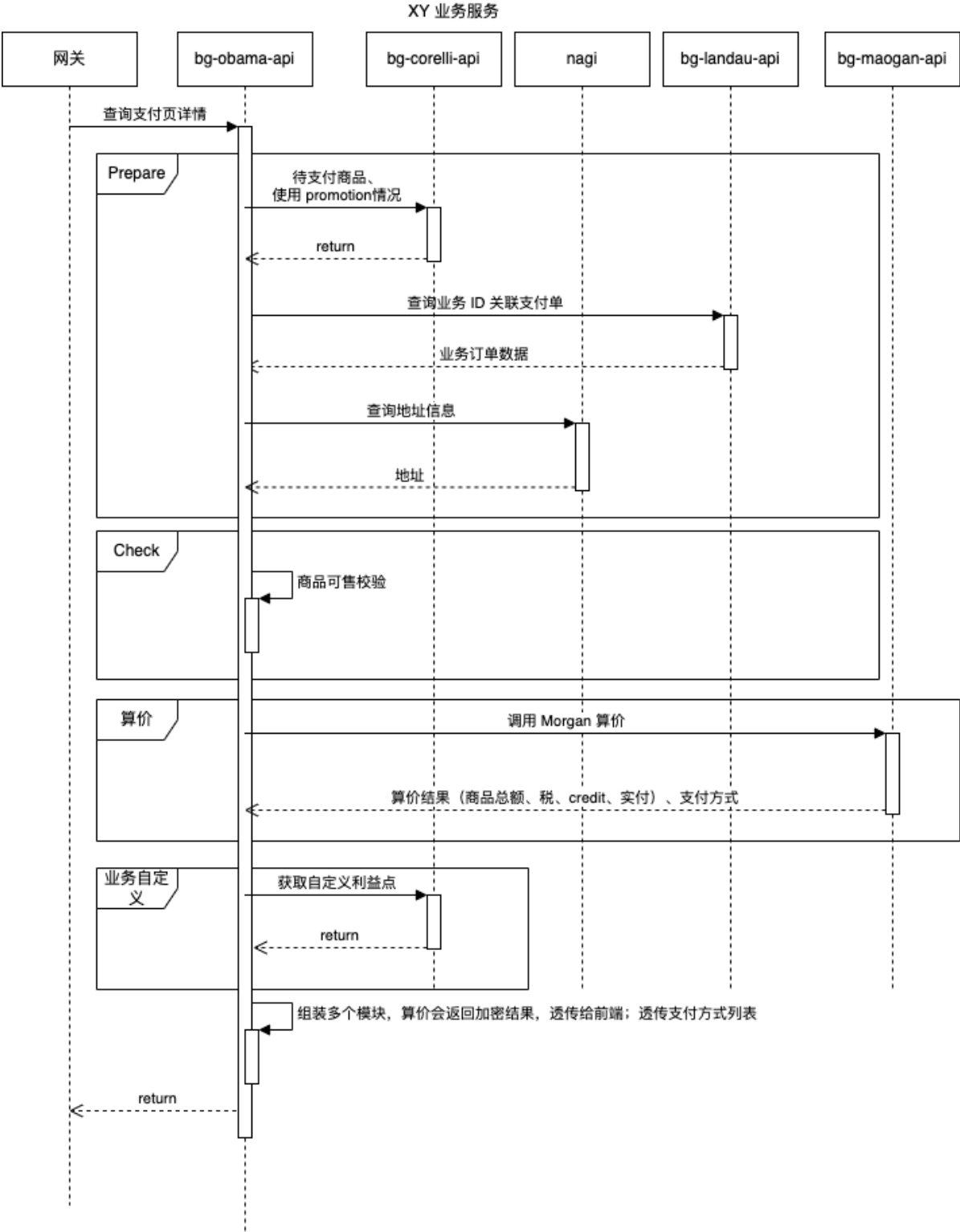








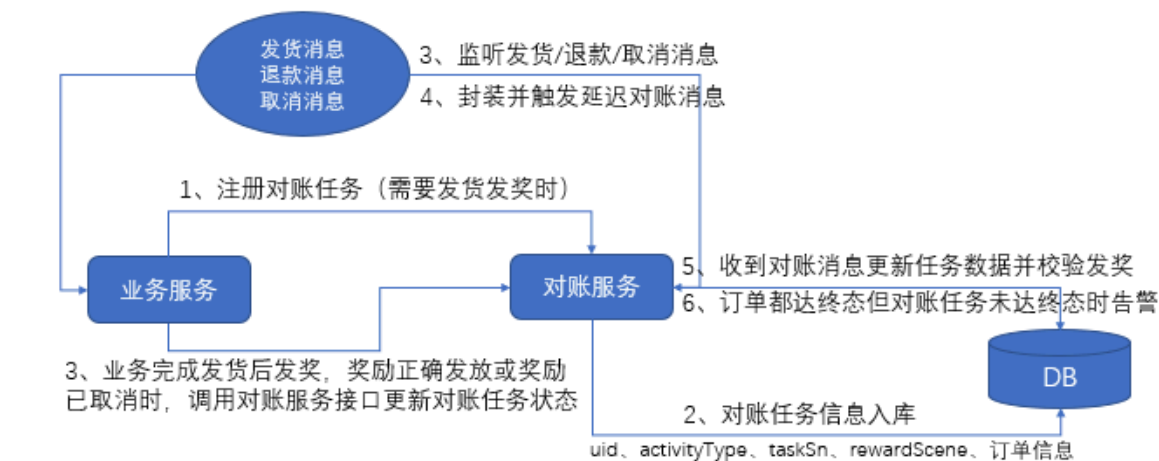




## 逆向

com.temu.api.bg.maggie.contract.service.spi.OrderRefundCheckActivitySpiService

## 赠品对账



## 置顶品

### 置顶品逻辑梳理

服务	方法	备注	操作
jones	com.temu.bg.jones.api.service.helper.TopGoodsHelper#checkGoodsAndSkuValid	本本品校验，敏感品校验，价格校验	合并
	com.temu.bg.jones.api.service.helper.TopGoodsHelper#validTopGoods	前台资格接口使用，无校验？要不要加	合并
	com.temu.bg.jones.api.service.utils.GoodsUtils#genPreInsertGoodsList()	获取goodsId, nz_goods, top_goods_list相关置顶品，无校验	合并
	com.temu.bg.jones.api.service.landpage.biz.LandPageNotLoginService#getTopGoodsStyle	未登录获取置顶品信息	合并
	com.temu.bg.jones.api.service.helper.GoodsQueryHelper#getTopPayGoods	获取置顶品信息，本本品校验，定制品校验 sm_op_goods>goods_id>cid	合并
	com.temu.bg.jones.api.service.utils.GoodsUtils#topGoodsList	获取nz_goods, top_goods_list，用来进行打点	合并
	com.temu.bg.jones.api.common.utils.UriUtils#genLandingUrl	跳链上拼接reques里的topGoods，用来落地页承接	合并
	com.temu.bg.jones.api.service.download.DownloadInfoCommand#genGoodsUrl	将置顶品拼到跳链上	合并
texas	com/temu/api/bg/texas/service/goods/context/ContextBuildHelper.java:297	优先从跳链获取cid前插，然后拼接request前插商品	删除
	com.temu.api.bg.texas.service.goods.impl.freebies.FreebiesPageGoods	置顶品校验，替换品逻辑	删除
	com.temu.api.bg.texas.service.goods.activity.AdChannelHelper#replaceGoods	替换品逻辑	迁移

大致可以分为：获取置顶品goodsId，校验置顶品，置顶品售罄替换

置顶品分为 付费品，售罄品（展示文案可能有区别），免费品

建议优先迁移texas逻辑，后续合并jones逻辑

置顶参数：goods\_id, sm\_op\_gid, nz\_goods, top\_goods\_list, \_x\_cid

## 统一发货

### 1.地址确认接口

path: /api/generic/jones/daily/gift/confirm/address

### 2.弹窗

DAILY\_GIFT\_SHIP\_TOGETHER\_PICK\_FREE\_GOODS

### 3.活动结束消息

```
{
  "uid": 1111111111,
  "bg_id": "abcde",
  "api_uid": "abcde",
  "installed_token": "abcde",
  "event_type": "DAILY_GIFT_ACTIVITY_END_TIME",
  "site": 102,
  "data": {
    "round_id": "R1111111111",
    "iris_dto": {}
  }
}
```

#### 4.创单重试

```
{
  "uid": 1111111111,
  "bg_id": "abcde",
  "api_uid": "abcde",
  "installed_token": "abcde",
  "event_type": "UNIFIED_SHIP_CREATE_ORDER",
  "site": 102,
  "data": {
    "round_id": "R1111111111",
    "iris_dto": {}
  }
}
```

#### 5.发券消息

```
{
  "uid": 1111111111,
  "bg_id": "abcde",
  "api_uid": "abcde",
  "installed_token": "abcde",
  "event_type": "UNIFIED_SHIP_SEND_COUPON",
  "site": 102,
  "data": {
    "coupon_amount": 199,
    "round_id": "R1111111111",
    "iris_dto": {}
  }
}
```

#### 6.延迟校验

```
{
  "uid": 1111111111,
  "bg_id": "abcde",
  "api_uid": "abcde",
  "installed_token": "abcde",
  "event_type": "DAILY_GIFT_ZERO_ORDER_DELAY_CHECK",
  "site": 102,
  "data": {
    "round_id": "R1111111111",
    "iris_dto": {}
  }
}
```

## 7.主接口

daily\_gift\_ship\_together\_info

[B] TVADS-850327 【赠品】分天领尝试团期结束统一发货

[【赠品】分天领尝试团期结束统一发货](#)

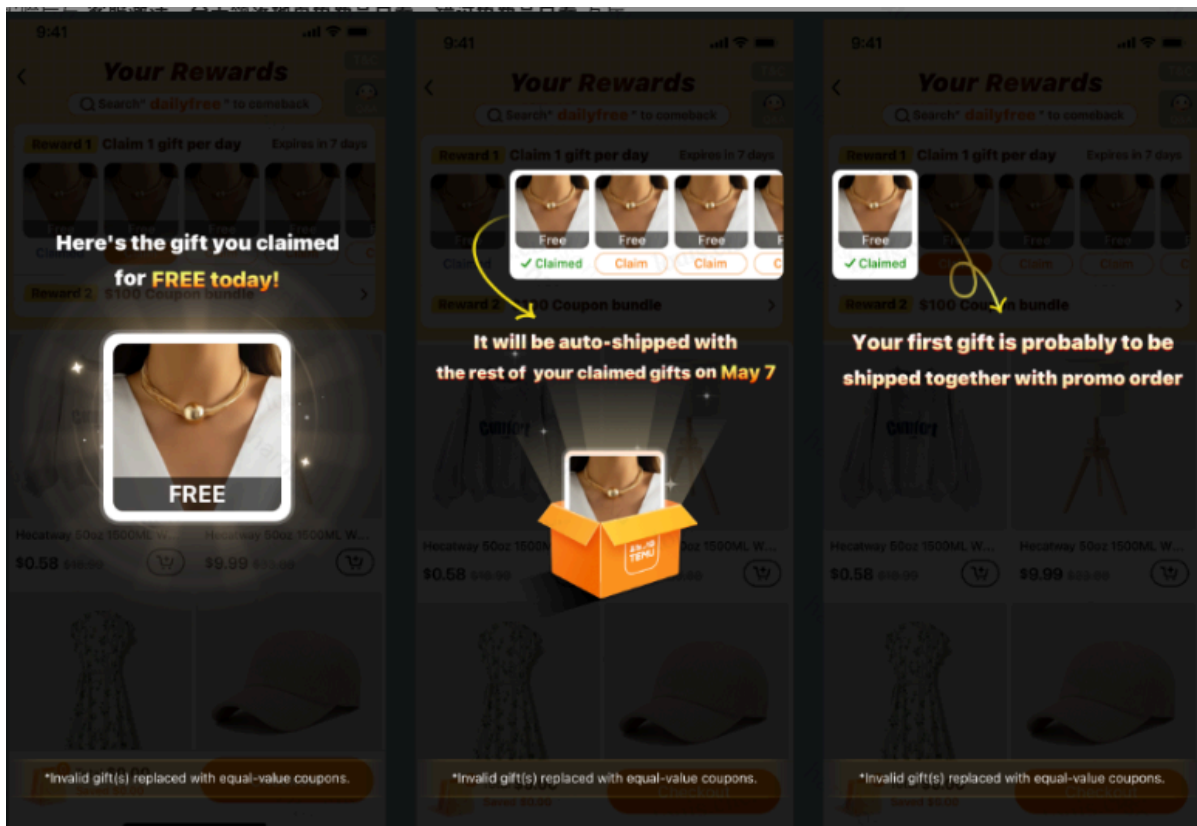
## 1.实验&落地页逻辑

- 实验时机，在**第二件免费品可领 & 进落地页 & 未进过实验 & 剩余可领\*\***免费品件数大于2\*\* 时 过实验，并更新bat
- 命中实验后与 **客服激活、分天领落地页免费品兑券、错过免费品兑券** 互斥
- 在可领取第二件免费品后前端出**地址模块**，用户确认地址，后端保存改地址，在发货时使用
- 落地页新增**事件列表**，展示用户当前所处状态，当曝光 DAILY\_GIFT\_SHIP\_TOGETHER\_PICK\_FREE\_GOODS弹窗后展示
- 在用户进实验后写一条**活\*\*动结束消息\*\***，数据量小，复用jones-freebies-delay-send-event 保存48h，延迟12h，通过循环消费消息来在分天领活动过期后自动创单+售罄品替换为发券
- 第二天领取免费品后**事件列表**为状态2，最后一天领完免费品后**事件列表**更新为状态3
- 后端新增地址确认接口，用来存储地址，在创单时使用
- 分天领挽留弹窗屏蔽GUIDE\_ORDER、GUIDE\_GRAB，加购弹窗屏蔽DAILY\_PICK\_FREE\_GOODS

## 2.弹窗

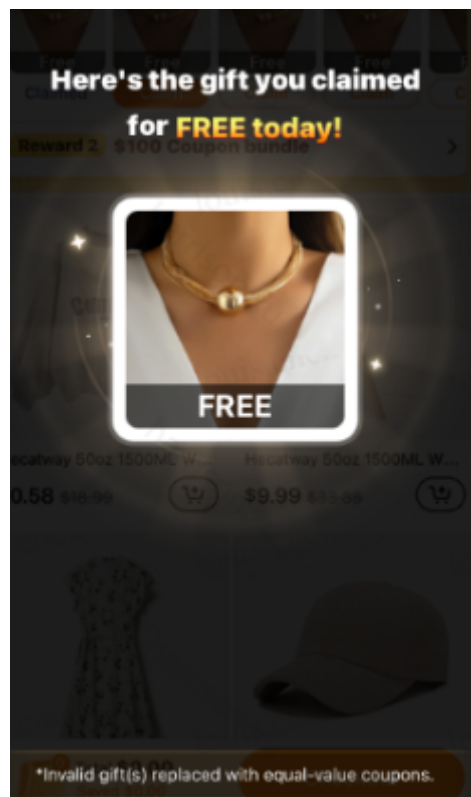
1.每次领取免费品后出

style: DAILY\_GIFT\_PICK\_GIFT



2.领第二件免费品后出

style: DAILY\_GIFT\_SHIP\_TOGETHER\_EDUCATION



3.最后一天领取后出

windowType: DAILY\_GIFT\_PREPARE\_SHIPPING



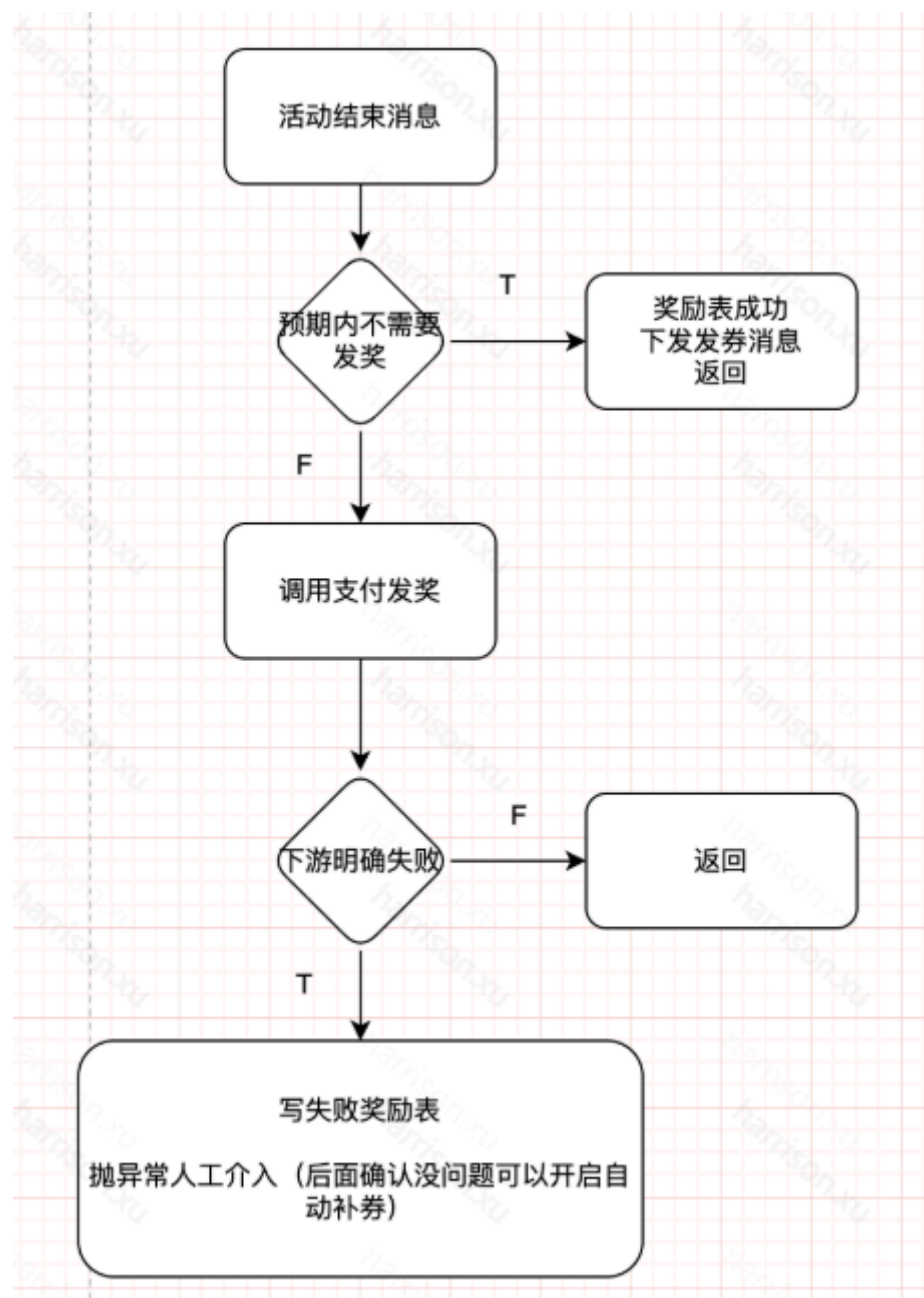


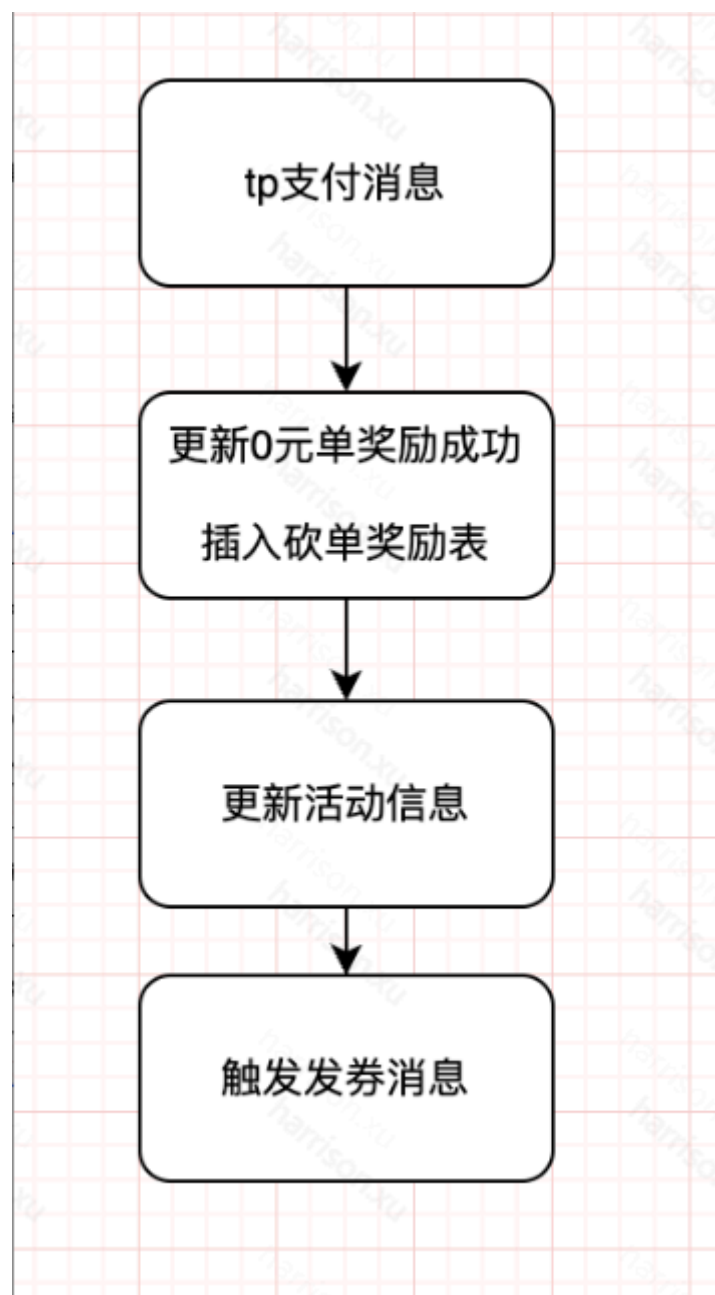
### 3.发券+创单

#### [0元单mock支付切创单并支付流程SOP](#)

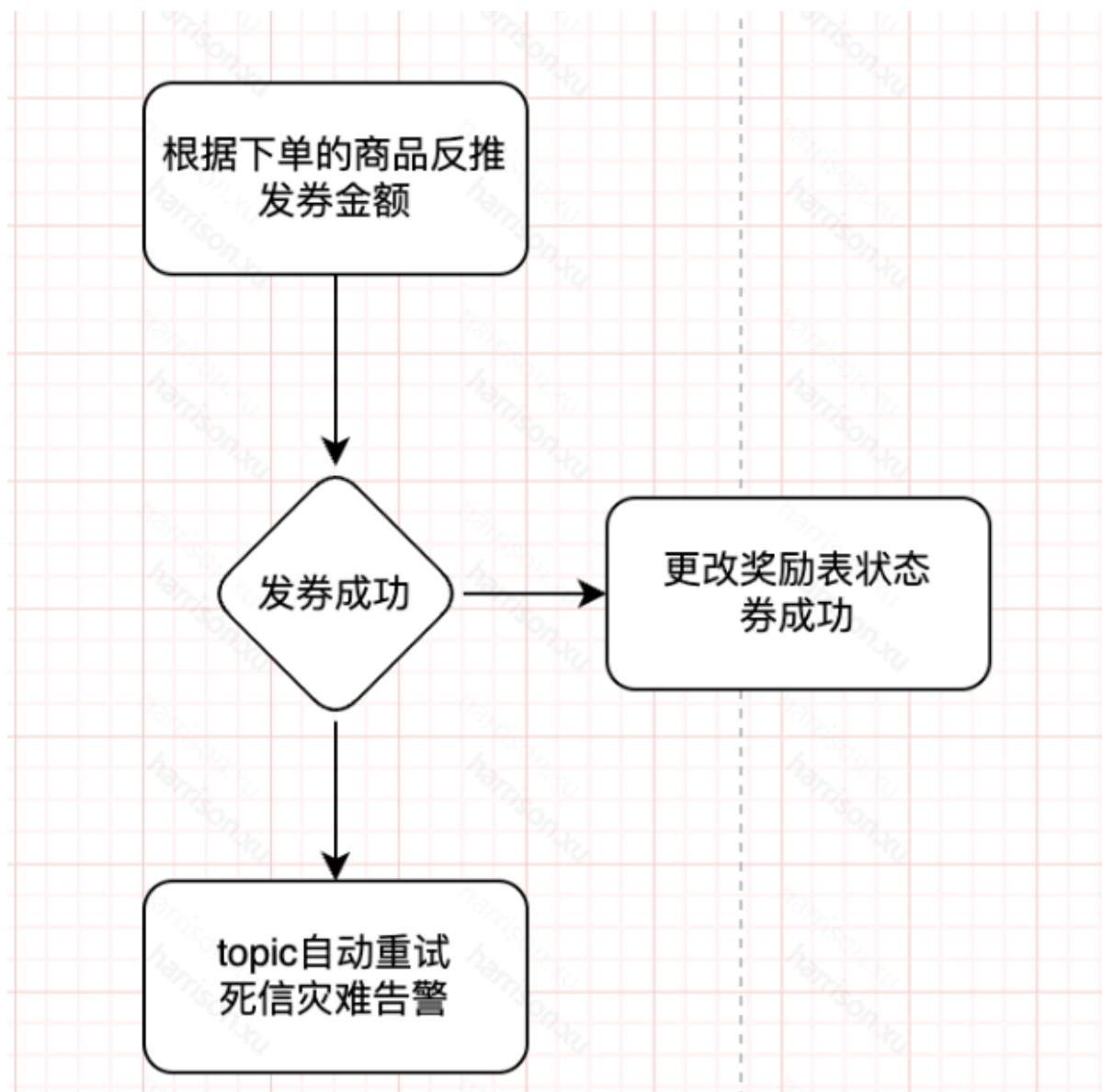
1. 通过消费活动结束时间来创单+发券，创单需要税优惠+邮费优惠
2. 活动非已完成不处理

0元单发奖

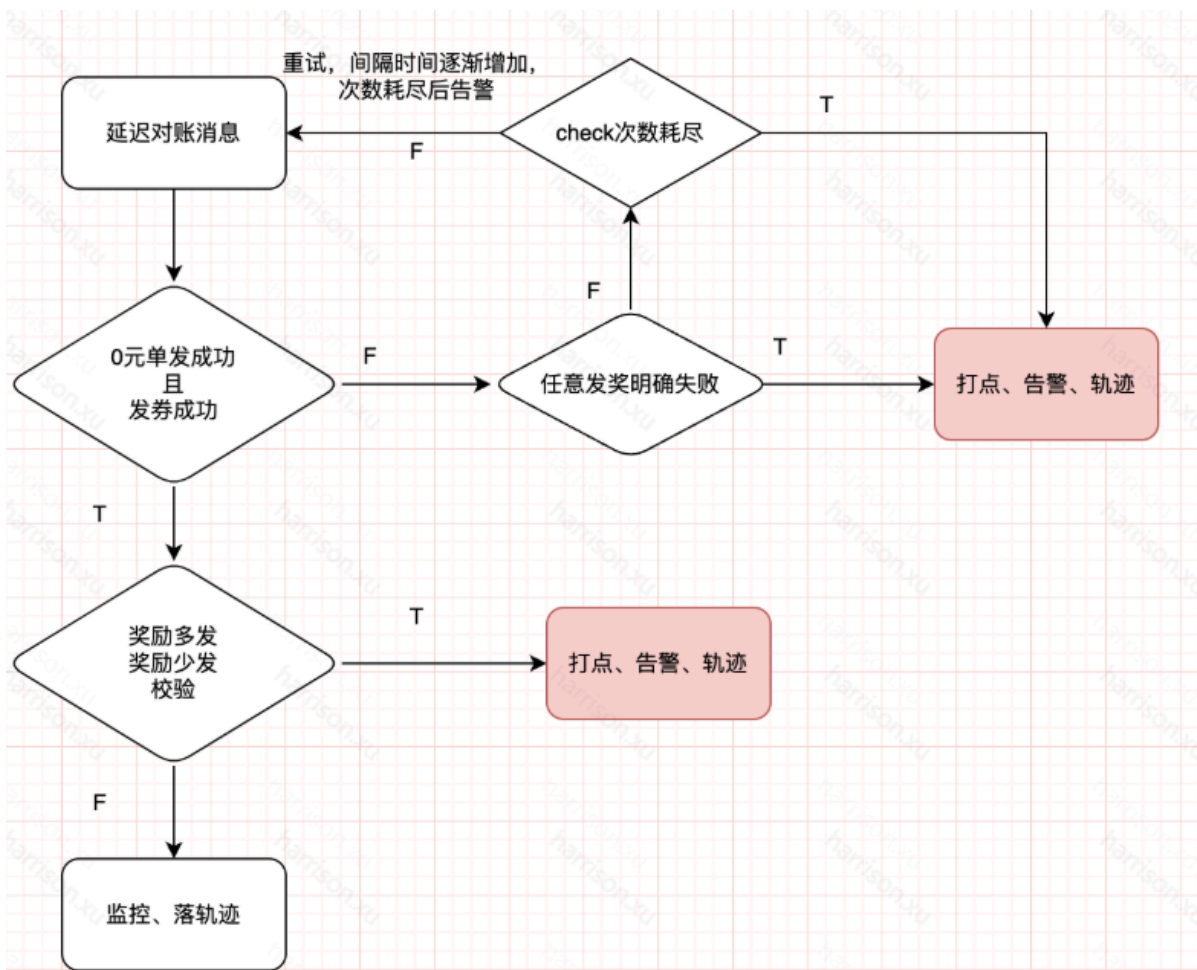




发券



延迟对账



## 4.客服平台适配

小流量不适配

### 砍单

- 1、兼容发奖前的砍单
- 2、发奖后的砍单 券回收、商品回收。 回收比例打点监控
- 3、用户在创单前砍单要对第一件免费品进行砍单

## 5.测试点

- 砍单：当用户退掉免费品后，已pick的免费品不再发货
- 发奖时首单已退单
- 发奖异常case，预期能及时感知
- 0元单rpc失败
- 0元单假成功
- 0元单成功、发券失败
- 失效商品补券逻辑
- 全部失效
- 部分失效
- 全部有效

## 6.对账

确保用户claim的商品一定创单+发券了

同一件品不能即发券又创单

不能多次创单