

Week 1 基础功能实现

week1中我们需要实现这个系统三端的基本功能：客户端、后端、管理端

推荐将整套系统托管在AWS的云服务器和RDS上。当然，如果你本身就有其它提供商的服务器，比如阿里云或者腾讯云，也可以使用。

客户端

客户端面向移动设备，使用嵌套的webview，不需要写完整的页面，只需要完成几个基础组件，以提供后端的对接能力：

- 商品列表
- 商品概览，sku选择
- 商品列表的filter
- 商品购买按钮

Filters ▾

Best Seller



Amazon Basics 48 Pack AA High-Performance Alkaline Batteries, 10-...

48 Count

★★★★★ 441,828

\$15³⁶ (\$0.32/Count)

\$14.59 with Subscribe & Save discount

50% off gift wrap service: code GIFTWRAP50

FREE Shipping to Hong Kong when you spend over \$49.00 on eligible items

Amazon's Choice



Amazon Basics 36 Pack AAA High-Performance Alkaline Batteries, 10-...

36 Count (Pack of 1)

★★★★★ 423,736

\$11²⁶ (\$0.31/Count)

\$10.70 with Subscribe & Save discount

50% off gift wrap service: code GIFTWRAP50

FREE Shipping to Hong Kong when you spend over \$49.00 on eligible items

Best Seller



Amazon Basics 8 Pack 9 Volt Performance All-Purpose Alkaline...

8 Count (Pack of 1)

★★★★★ 147,683

\$11⁹⁹ (\$1.50/Count)

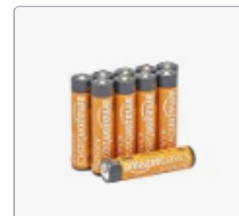
\$11.39 with Subscribe & Save discount

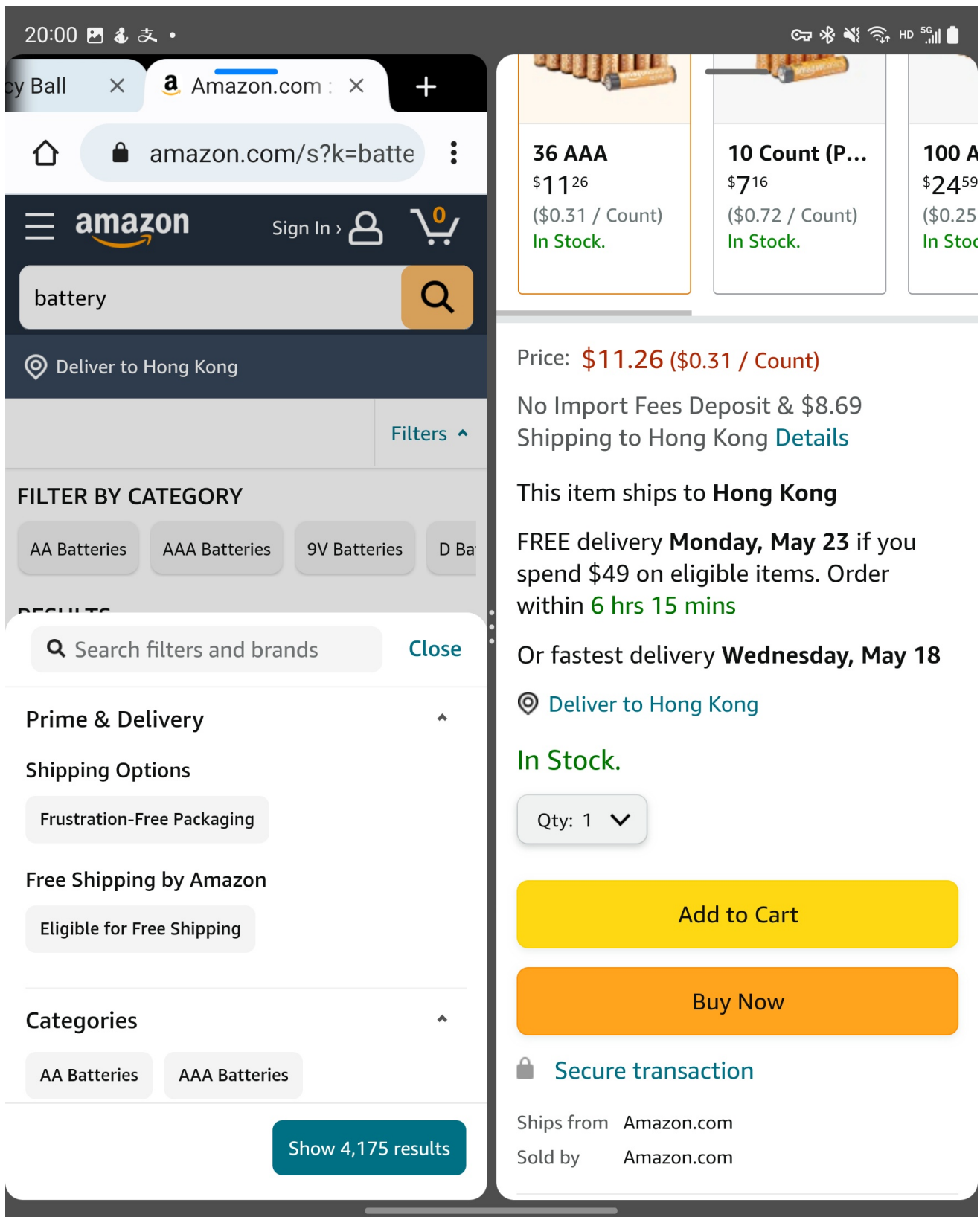
Visit the Amazon Basics Store

★★★★★ 423,736

Amazon Basics 36 Pack AAA High-Performance Alkaline Batteries, 10-Year Shelf Life, Easy to Open Value Pack

Amazon's Choice for "battery"

Pattern Name: **Battery**Style: **36 AAA**



成功或者失败以Toast的形式返回。

需要有3种状况：

1. 订单提交成功，购买完成
2. 订单提交失败，优惠券无效
3. 订单提交失败，商品无货
4. 订单提交失败，其他原因

后端

后端可以采用Mybatis-plus等辅助框架，登陆采用OAuth2授权码 + JWT方案，我们将用JWT来模拟购买和消费COUPON的用户校验场景。这套简易系统暂时不需要考虑多端挤占问题。

后端要求实现对**COUPON**表、优惠活动表的和用户表、商品表、订单表CRUD，同时用户具备简单的两种权限，user和admin，admin可以登录到管理后台以及生成COUPON，而user不允许。

数据库选型Mysql，第一阶段后端直接使用JDBC对数据库进行操作。

COUPON表需要具备以下字段：

字段名	含义
id	随机生成的券码序列号
status	未使用或者已使用
type	表示是满减、固定金额折扣还是倍率折扣
discount	折扣金额，如果是满减和固定金额，则填写数额；如果是倍率折扣则填写小于1的浮点数
condition	满减价格
activeTime	生效时间，如果未达到则不可用
range	['mid','mid'] 用于表明COUPON生效商品
expiredTime	过期时间

用户表请参考主流的开源电商/外卖平台模型。

优惠活动表与商品表关联，用于记录在活动期间直接折扣而不是通过发放COUPON实现优惠的商品，前端在访问商品价格数据的时候，后端应当优先查询该表获取价格信息，并且在生成订单信息的时候应当使用该表中的商品价格数据。

管理端

管理端为方便搭建，建议采用开源的admin框架，实现对券发放的管理，日志查看以及用户列表、商品列表管理。

日志系统需要记录用户的请求和结果、来源IP、数据库对应操作的返回值。

业务逻辑

要求达成，admin通过登录后台能操作管理批量生成COUPON，用户登录前端后能够通过后端获取用于测试的优惠券列表。

同时，具备模拟购买流程，用户可以在任何时间进行常规购买，或者在优惠有效期带coupon提交订单，后端需要校验coupon并且完成折扣和购买。

若成功，将该订单的信息插入订单表

若失败，原因回传到前端后以Toast显示。

抢购/点击记录能够在管理端的日志页看到，同时管理端的用户页需要能够显示用户以及抢购过多少数量的商品。

思考

高并发的时候如何将库存与抢购请求处理分离，以降低数据库性能开销？

有没有一种降低系统耦合度的优惠价格应用方式？比如说优惠表由一个监控服务维护，到优惠时间后由维护服务自动更改商品表的价格，而不用在购买和查询的时候需要后端同时与两个数据库表进行交互。