

# 互联网应用开发技术-迭代 2 细则

## 1. 迭代要求-前端&后端通信

请你使用 React、React-Router(或相关框架)和 Ant Design(或相关框架)来开发你的在线书店的前端工程，使用 Spring Boot 开发你的在线书店的后端工程，并使用 fetch API 实现前后端工程的通信。具体应该满足以下要求：

- A. 前端：你应该在迭代一的基础上，整合几次平时作业的内容，将开发的前端页面互相联动起来，例如将书籍放入购物车后就可以在购物车中看到相应的书籍；下订单后在订单列表页就可以看到相应的订单。
- B. 后端：能够响应前端发送的[获取图书列表](#)、[获取图书详情](#)、[获取订单列表](#)等功能，并且返回用户需要的数据或处理结果。如果时间充裕，完成后端响应前端发送的[放入购物车](#)、[下订单请求](#)，同样返回用户需求的数据或处理结果。关于购物车的实现，有些同学认为应该是纯前端的对象，有些同学认为是在后端内存中的对象，还有些同学认为是应该存储数据库的对象。在实际开发中，建议采取最后一种方式，因为购物网站通常的用户体验是购物车信息在持久化存储的，即用户即使登出网站，下次登录时购物车数据依然存在。
- C. 前后端通信：采用 Ajax 异步通信方式，前端展示的数据都通过向后端请求获得，后端从 MySQL 数据库中读取数据，并以 JSON 方式发送给前端。
- D. 数据库访问：后端使用 Spring JDBC Data 方式访问数据库，如果你已经使用了 Spring JPA 访问数据库，则无需修改为使用 Spring JDBC Data 访问数据库。
- E. 数据库设计：库结构和表结构合理，尤其注意上课提到的几个避坑问题。
- F. 总体而言，迭代二是前后端贯通的系统，前端展示的数据均来自于后端的数据库中，用户在前端执行的操作也需要反应到后端的数据库中。

## 2. 提交要求

- A. 请提交你的 React 工程的源代码，以及 Spring Boot 源代码。注意：前端勿压缩提交 node\_module 子目录。后端勿压缩提交 lib 子目录。
- B. 提交你的数据库设计的 SQL 脚本。

### 3. 评分标准：共 20 分

#### A. 功能（4 分）

- i. 前端页面互相之间产生联动，将书籍放入购物车后就可以在购物车中看到相应的书籍；下订单后在订单列表页就可以看到相应的订单。2 分
- ii. 前端展示数据来自于后端访问数据库得到的结果。2 分

#### B. 设计方案（8 分）

- i. 前端工程结构合理，分为 Component、Service、View、Util 等包进行管理，你不一定完全按照结构规划，可以根据自己的设计进行定制裁剪，但需要逐一解释设计理由。2 分
- ii. 后端工程结构合理，目前后端设计要点尚未展开，但至少应该关注将实体类和 Controller 类分离开，放在不同的包中。关于实体类，参见“D. 数据库访问”中的描述。2 分
- iii. 前后端通过异步方式通信，并且前后端传递 JSON 数据，JSON 数据格式设计合理。2 分
- iv. 请你对照你的代码详述从前端 JavaScript 发出请求->后端 Java 处理->访问数据库->后端 Java 抽象和组装数据并发送回前端->前端刷新页面展示的全过程。2 分

#### C. 数据库设计（4 分）

- i. 库结构和表结构设计合理，例如，Order 和 OrderItem 外键关联，以及 User 和 UserAuth 外键关联正确。2 分
- ii. 请你详述你的数据库设计方案，包括主键类型、字段类型、字段长度、字段约束等设计依据。2 分

#### D. 数据库访问（4 分）

- i. 使用 Spring Data JDBC 或 JPA 正确地连接并访问数据库。能够解释建立数据库连接和数据持久化操作的完整处理过程。2 分

- ii. 使用 Spring Data JDBC 访问数据库时，正确地编写 SQL 语句实现访问数据库逻辑；使用 Spring JPA 访问数据库时，正确地使用 Repository 中的方法，或者正确编写定制的持久化方法。1 分
- iii. 无论使用何种方式访问，都将数据库中的数据抽象成实体类，然后再做处理，这样在后续迭代中可以持续改进。1 分