

# 应用系统体系架构 — 作业 1

## 1. 作业要求

请你在大二开发的 E-BookStore 系统的基础上，完成下列任务：

- A. 请你设计一个 Service，它包含一个计时器，可以记录用户每次登录 E-BookStore 后的会话保持的时间，具体要求为：
  - i. 在你设计的负责用户登录和登出的 Controller 中，在登录方法(login)被调用时，调用这个定时器 Service，在其中初始化计时器，并开始计时；
  - ii. 在登出方法(logout)被调用时，调用这个定时器 Service，在其中停止计时器，并获取计时器的计时值，返回给 logout，logout 方法会返回给前端显示这个时间；
  - iii. 如果你之前的系统未针对用户登录登出专门编写 Controller，那么就按照上述要求新编写一个 Controller 以满足要求；
  - iv. 在编写代码时，请正确配置 Controller 和 Service 的@Scope 属性；
  - v. 请你使用多个浏览器同时登录你的系统，并且在登出时观察它们获得的计数值，并编写一个文档，解释你配置的@Scope 属性值的依据，并对最后得到的计时值截图贴在文档中，将文档与代码一起上传。
- B. 针对现有 E-BookStore 中下订单功能，进行事务控制，具体要求为：
  - i. 假设你的代码中在下订单的 Service 中，需要完成两件事情，在数据库的 Order 表中插入一条记录，在 OrderItem 表中插入多条记录。例如，如果一个订单包含 3 种不同的书，那么在 Order 表中会插入一条记录，在 OrderItem 表中会插入 3 条记录；
  - ii. 按照上述操作，Service 需要调用 OrderDao 和 OrderItemDao 两个对象的响应方法来实现下订单，而且必须保证两个表的数据要么都插入成功，要么都不插入，即这两个表的插入操作必须在一个事务中完成；

- iii. 如果你之前的系统不是这样实现的，那么就按照上述要求重构你的代码以满足要求；
- iv. 请你参照上课讲解的 transfer 样例，通过声明方式实现上述事务管理功能，并按照课件中第 25 页所列表格通过修改@Transactional 中的事务传播属性值来观察不同的执行结果，在文档中记录下结果，并解释为什么会出现这样的结果，**将文档与代码一起上传。**

## 2. 提交要求

请将你自己编写的源码、脚本和文档压缩后上传，勿压缩整个工程提交，尤其是不要压缩第三方的 Jar 包。

## 3. 评分标准：

- A. 正确完成“作业要求 A”中的功能（1.5 分）
- B. 正确解释所使用的 Scope 属性值的原因（0.5 分）
- C. 正确完成“作业要求 B”中的功能（2 分）
- D. 正确解释所使用的 Transactional 传播属性值的原因（1 分）
- E. 如果不能将作业要求 A 或 B 整合在你的 E-BookStore 中，那么请开发单独的针对作业要求完成的独立工程并提交，但是需要扣分，每项未整合到系统中的作业要求扣 1 分。