2024-09-20

应用系统体系架构 — 作业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：

## 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：

### 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；

### 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；

### 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；

### 在编写代码时，请正确配置Controller和Service的@Scope属性；

### 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。

## 参照上课给的样例，编写通过Kafka消息中间件处理订单的功能，具体要求为：

### 前端发送的下清单请求被后端的Controller接收，然后按照你设计的消息格式，组装下订单消息，并将其发送到Kafka中你预先创建好的Topic中；

### 在后端增加类似样例中用@Component注解的消息监听器类，监听指定Topic中的下订单消息；

### 在监听到下订单消息后，监听器类会调用下订单服务完成下订单，并且将下订单的处理结果发送到预先创建好的另一个Topic中；其中，下订单服务应该与同步处理下订单功能复用相同的Service，以提高代码的可维护性；

### 本次作业不需要实现下订单后前端展示下订单成功的页面，在后端控制台输出，编写文档，将前端发送请求->后端接收到请求进行处理->改写数据成功的流程截图，并对流程进行文字说明，以证明你开发的异步接收订单进行处理的功能正常实现了。

# 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。

## 请使用第00单元中的“作业模版docx”作答，并将文件名改为“学号-作业X.docx”，其中X为作业序号。

# 评分标准：

## 正确完成“作业要求A”中的功能（2分）

## 正确解释所使用的Scope属性值的原因（0.5分）

## 正确完成异步“下订单”的功能，即后端能够使用Kafka消息中间件，在接收到前端发送的下订单消息后，调用你之前的下订单服务生成订单，并将其持久化到数据库中（2.5分）

## 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。