202X-XX-XX

应用系统体系架构 — 作业1

学号：521030990006

姓名：VAHAGN GHAZARYAN

得分：

# 本次作业回答如下：

## Explain the basis for the @Scope attribute value you configured

## **TimerService with Session Scope:** I set the TimerService to have a **session** scope so that each HTTP session has its own unique instance of the TimerService. This ensures that each user session will have an isolated timer, preventing any interference between different users/browsers.

## 

## **AuthController with Default (Singleton) Scope:** The controller is the default **singleton** scope because it is stateless and doesn't maintain user-specific data. A single instance of the controller is efficient enough to handle multiple requests.

## If I have chosen to use a **session** scope for the AuthController, it means a new instance of the controller is created for each HTTP session. **This could be useful if the controller needs to keep some state that is specific to the user session, such as user preferences**. However, this could also consume more resources compared to a singleton scope, so I don’t use it for now.

## 

## **DEMO:**

## 

## 

## 

## Asynchronous "ordering" function, the backend use the Kafka message middleware.

## Front end sends post request to **http://localhost:8080/orders**

## **OrderController** receives POST request and uses KafkaProducer to send a topic (*topic name = order-topic*)

## **KafkaProducer** takes the order and pushes it to Kafka.

## **KafkaConsumer** listens to the topics. When it gets a new order, it processes it (calling **OrderService.addOrder()**) and sends the result back to another topic (*topic name = order-result-topic*)

## **KafkaConsumer** listens to the result topic **(ListenOrderResultTopic method) and logs the successful operation**

## **KafkaTopicConfig** defines the Kafka topics.

## **Why it's Async?**

## OrderController doesn't wait. It sends the order and moves on. KafkaConsumer works separately, picks up the order, and processes it.

## **CODE**

## 

## 

## 

## 

## 

## 

## 

## **DEMO**

## 

## 