ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | П.А.Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7 |
| «Разработка приложения с асинхронной очередью сообщений» |
| по курсу: Технологии разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4831 |  | 19.11.2020 |  | К.А.Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

1. **Цель работы**

Целью работы является реализация простой системы распределенной репликации (“писателичитатели”). Скачайте и разверните Apache Kafka.Модифицируйте свое приложение со встраиваемой базой данных так, чтобы его можно было запустить в нескольких экземплярах на разных портах. Реализуйте в рамках своего приложения Producer и Consumer такие, что Producer при каждой операции записи оповещает соответствующий топик Consumer при получении информации из топика записывает обновление в локальную (встроенную в приложение) базу. Продемонстрируйте, что информация, записанная одним приложением, доступна второму приложению.

1. **Вариант задания**

Коллекционирование (нумизматика, филателия и пр).

1. **Код программы**

@Service  
@Log4j  
public class KafkaConsumer {  
  
 @Autowired  
 MoneyRepository moneyRepository;  
  
 @KafkaListener(topics = "TEST2", groupId = "group0")  
 public void consume(ConsumerRecord<String,String> message) throws IOException {  
 System.out.print(String.format("#### -> Consumed message -> %s", message));  
 ObjectMapper mapper = new ObjectMapper();  
 Money money = mapper.readValue(message.value(), Money.class);  
 moneyRepository.save(money);  
 }  
  
}

@Service  
public class KafkaProducer {  
  
 private static final String TOPIC = "TEST4";  
  
 @Autowired  
 private KafkaTemplate<String, String> kafkaTemplate;  
  
 public void sendMessage(Money money) throws JsonProcessingException {  
 ObjectMapper mapper = new ObjectMapper();  
 String message = mapper.writeValueAsString(money);  
 this.kafkaTemplate.send(TOPIC, message);  
 }  
}

1. **Результаты работы программы**

****

1. **Вывод**

В ходе выполнения лабораторной работы было разработано приложение с асинхронной очередью.