编号：

2020 - 2021学年第 二 学期



**实 验 报 告**

实验课程名称 软件架构与应用开发

专 业 班 级 计算机1803

学 生 姓 号 31801150

学 生 姓 名 张帅

实验指导教师 彭彬

一. 实验目的和要求

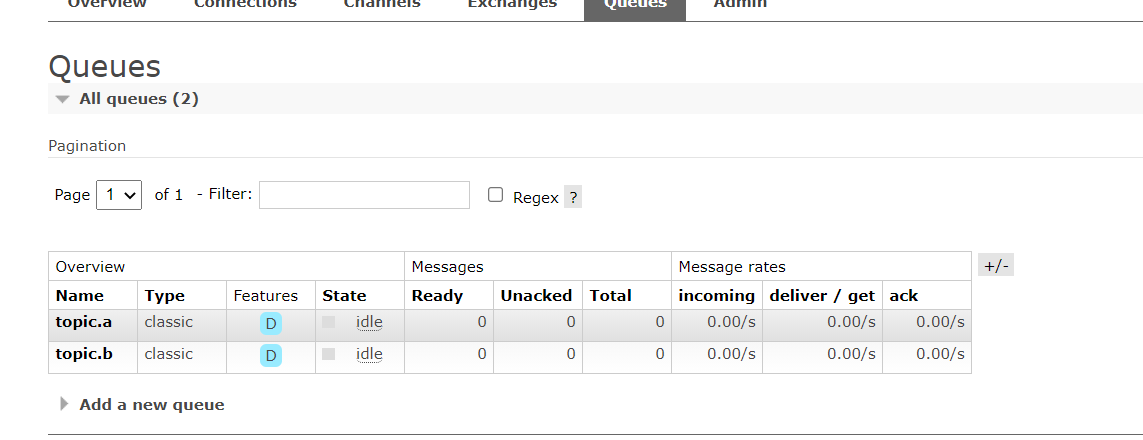
1.学习本节示例L11Rabbitmq\_TopicDemo

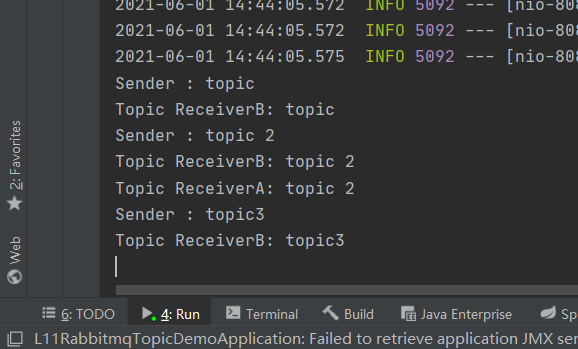
2.利用Rabbit MQ技术实现如下功能：

设计一个社团纳新功能，当确认一位新社团成员时，向原先的所有成员发送该成员的信息（学号、姓名、性别等）

二. 实验内容

1.学习本节示例L11Rabbitmq\_TopicDemo





2.利用Rabbit MQ技术实现如下功能：

设计一个社团纳新功能，当确认一位新社团成员时，向原先的所有成员发送该成员的信息（学号、姓名、性别等）

@Component  
@RabbitListener(queues = "topic.a")  
public class TopicReceiverA {  
 @Autowired  
 StuMapper stuMapper;  
  
  
 @RabbitHandler  
 public void process(String msg) {  
 ArrayList<Stu> re = (ArrayList<Stu>) stuMapper.queryAll();  
 for(int i=0;i<re.size();i++){  
 System.*out*.println("send to "+re.get(i).getName()+" "+msg);  
  
 }  
  
 }  
  
}

@Component  
public class TopicSender {  
  
 @Autowired  
 private AmqpTemplate amqpTemplate;  
  
 public void send() {  
 String context = "topic";  
 System.*out*.println("Sender : " + context);  
 this.amqpTemplate.convertAndSend("topicExchange", "topic.1", context);  
 }  
  
 public void send2(String msg) {  
  
 System.*out*.println("Sender : " + msg);  
 this.amqpTemplate.convertAndSend("topicExchange", "topic.a", msg);  
 }  
  
 public void send3() {  
 String context = "topic3";  
 System.*out*.println("Sender : " + context);  
 this.amqpTemplate.convertAndSend("topicExchange", "topic.b", context);  
 }  
  
}

@RestController  
public class TopicSendController {  
 @Autowired  
 private TopicSender sender;  
  
 @GetMapping("topicsend")  
 public void topic() throws Exception {  
 sender.send();  
 }  
  
  
  
 @GetMapping("topicsend3")  
 public void topic2() throws Exception {  
 sender.send3();  
 }  
}

@Configuration  
public class RabbitmqConfig {  
 @Bean  
 public Queue queueMessage() {  
 return new Queue("topic.a");  
 }  
  
 @Bean  
 public Queue queueMessages() {  
 return new Queue("topic.b");  
 }  
  
 @Bean  
 TopicExchange exchange() {  
 return new TopicExchange("topicExchange");  
 }  
  
 @Bean  
 Binding bindingExchangeMessage(Queue queueMessage, TopicExchange exchange) {  
 return BindingBuilder.*bind*(queueMessage).to(exchange).with("topic.a");  
 }  
  
 @Bean  
 Binding bindingExchangeMessages(Queue queueMessages, TopicExchange exchange) {  
 return BindingBuilder.*bind*(queueMessages).to(exchange).with("topic.#");  
 }  
}

