

廈門大學



软件学院

《中间件技术》实验报告

题 目 中间件技术实验四

姓 名 陈澄

学 号 32420212202930

班 级 软工三班

实验时间 2024/04/15

2024 年 04 月 15 日

1 实验目的

理解和应用服务服务器，能够基于应用服务器进行软件的组件式开发，能够从无到有使用中间件服务器的 Services 和 Facilities。

2 实验题目

单次小实验形式

使用消息中间件 Activemq 实现定时发送系统的处理器存储器使用情况的 Facility。

3 实验环境

系统：Windows11

编程语言：java

程序架构：springboot

4 实验步骤

1. 引入相关依赖

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.apache.activemq</groupId>
  <artifactId>activemq-client</artifactId>
  <version>6.0.1</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>5.3.13</version>
</dependency>
```

2. 编写定时发送消息程序（客户端）

使用 Spring 框架所提供的 `@Scheduled` 实现任务的定时，可以使用 `fixedRate` 字段给出特定的时间间隔（单位 ms）也可以使用 `cron` 字段给出特定的 `cron` 表达式。如：

“0 0 12 * * ?” 每天中午 12 点执行。

```
@Scheduled(fixedRate = 5000) // 每5秒发送一次数据
// 或者@Scheduled(cron = "0 0 12 * * ?")每天中午12点执行
public void sendDataToClients() {
    try {
        //与ActiveMq建立连接
        ConnectionFactory connectionFactory = new ActiveMQConnectionFactory(BROKER_URL);
        Connection connection = connectionFactory.createConnection();
        connection.setClientID(USER_NAME);
        connection.start();
        //设定消息确认模式（此处是自动确认）和TOPIC
        Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        Topic topic = session.createTopic(CHAT_TOPIC);
        MessageProducer producer = session.createProducer(topic);
        //发送文本消息
        String messageText = generateData();
        TextMessage textMessage = session.createTextMessage(s: USER_NAME + ": " + messageText);
        producer.send(textMessage);
        connection.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3. 编写获取当前系统和处理机状态方法（客户端）

使用 `OperatingSystemMXBean` 类获取有关操作系统的信息，该类也提供了一组用于监视和管理 Java 虚拟机所运行的操作系统的方法。

包括：

`getAvailableProcessors()`获取处理机数目

`getSystemCpuLoad()`获取 CPU 占用

使用 `Runtime` 类获取当前系统的占有内存，总内存和空闲内存

```

private String generateData() {
    String s = "";
    OperatingSystemMXBean osBean = (OperatingSystemMXBean) ManagementFactory.getOperatingSystemMXBean();

    // 获取系统处理器信息
    s += "系统处理器数目: " + osBean.getAvailableProcessors() + "\n";
    s += "系统处理机占用: " + osBean.getSystemCpuLoad() * 100 + "%" + "\n";

    // 获取系统内存信息
    long totalMemorySize = Runtime.getRuntime().totalMemory(); // 总内存
    long freeMemorySize = Runtime.getRuntime().freeMemory(); // 空闲内存
    long maxMemorySize = Runtime.getRuntime().maxMemory(); // 最大内存
    s += "总内存: " + totalMemorySize / (1024 * 1024) + " MB" + "\n";
    s += "空闲内存: " + freeMemorySize / (1024 * 1024) + " MB" + "\n";
    s += "最大内存: " + maxMemorySize / (1024 * 1024) + " MB" + "\n";
    // 模拟生成数据的逻辑
    return s;
}

```

4. 服务端创建消息监听并通过消息队列获取来自客户端的定时处理机和系统信息。

```

try {
    //与ActiveMq建立连接
    ConnectionFactory connectionFactory = new ActiveMQConnectionFactory(BROKER_URL);
    Connection connection = connectionFactory.createConnection();
    connection.setClientID(USER_NAME);
    connection.start();

    //设定消息确认模式（此处是自动确认）和TOPIC
    Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
    Topic topic = session.createTopic(CHAT_TOPIC);

    //创建消息监听事件，便于收到其他用户消息
    MessageConsumer consumer = session.createDurableSubscriber(topic, USER_NAME);
    consumer.setMessageListener(message -> {
        String receivedMessage = null;
        try {
            receivedMessage = ((TextMessage) message).getText();
        } catch (JMSEException e) {
            throw new RuntimeException(e);
        }
        System.out.println("Received: " + receivedMessage);
    });
}
catch (Exception e){
    e.printStackTrace();
}
}

```

5 实验结果

```
↑ 2024-04-22T15:27:54.021+08:00 INFO 207
↓ 2024-04-22T15:27:54.024+08:00 INFO 207
↔ 2024-04-22T15:27:54.705+08:00 INFO 207
⇅ Received: user2: 系统处理器数目: 12
   系统处理机占用: 0.0%
   总内存: 48 MB
   空闲内存: 21 MB
   最大内存: 4074 MB

Received: user2: 系统处理器数目: 12
系统处理机占用: 1.7744651432101466%
总内存: 48 MB
空闲内存: 20 MB
最大内存: 4074 MB

进程已结束, 退出代码为 130
```