



JavaEE 实验七

实验报告

姓　　名　　 Mcrivers

班　　级　　 软件工程2102班

学　　号

提交日期　　 2024.1.7

目录

[一、基础实验——无状态会话 Bean 的调用 1](#_Toc155530736)

[1. 运行结果截图 1](#_Toc155530737)

[2. 修改后的关键代码，及相应的运行结果或报错信息 1](#_Toc155530738)

[3. 根据实验过程，总结客户端程序调用EJB的过程 3](#_Toc155530739)

[4. 碰到的问题及解决方案或思考 3](#_Toc155530740)

[5. 实验收获及总结 3](#_Toc155530741)

## 一、基础实验——无状态会话 Bean 的调用

## 1. 运行结果截图

**5、启动服务器，然后运行LoginClient.java，观察输出结果**

观察到，控制台输出了“true”，代表登录成功

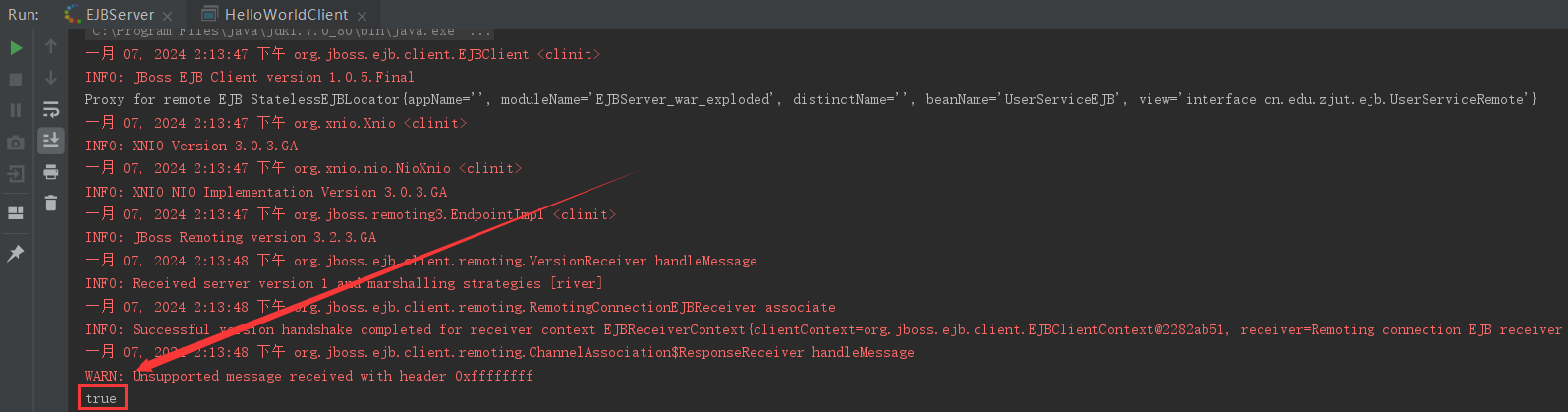


图1.1 成功登录的控制台输出截图

**6. 修改登录用户名和密码，观察输出结果**

观察到，控制台输出了“true”，代表登录成功



图1.2 登录失败的控制台输出截图

**7. 修改ejb-project1工程的用户登录逻辑，重新部署，运行LoginClient.java， 观察输出结果**

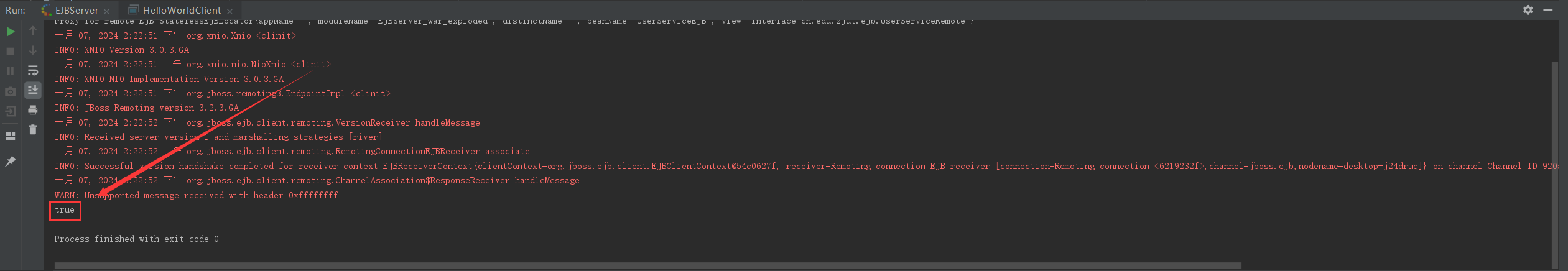


图1.3 成功登录的控制台输出截图

## 2. 修改后的关键代码，及相应的运行结果或报错信息

@Stateless(name = "UserServiceEJB")  
public class UserServiceBean implements UserServiceRemote {  
 private static final String *CORRECT\_USERNAME* = "zjut";   
 private final String CORRECT\_PASSWORD\_HASH;  
  
 public UserServiceBean() {  
 this.CORRECT\_PASSWORD\_HASH = hashPassword("zjut");  
 }  
  
 @Override  
 public boolean login(String username, String password) {  
 if (username.equals(*CORRECT\_USERNAME*) && validatePassword(password)) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 private boolean validatePassword(String password) {  
 String hashedPassword = hashPassword(password);  
 return hashedPassword.equals(CORRECT\_PASSWORD\_HASH);  
 }  
  
 private String hashPassword(String password) {  
 try {  
 MessageDigest md = MessageDigest.*getInstance*("SHA-256");  
 byte[] hashedBytes = md.digest(password.getBytes());  
  
 StringBuilder stringBuilder = new StringBuilder();  
 for (byte b : hashedBytes) {  
 stringBuilder.append(String.*format*("%02x", b));  
 }  
 return stringBuilder.toString();  
 } catch (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
}

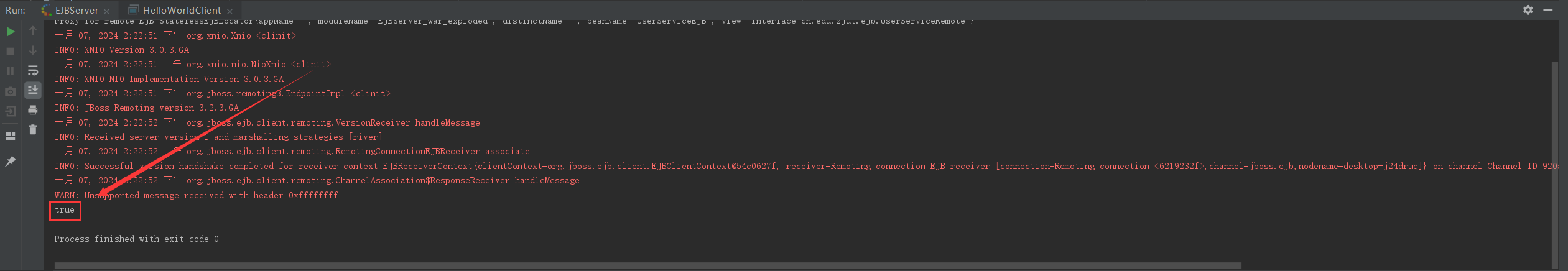


图1.4 运行结果

## 3. 根据实验过程，总结客户端程序调用EJB的过程

1. **设置 JNDI 属性：** 在客户端代码中，设置 JNDI 属性。
2. **创建 InitialContext：** 使用设置的 JNDI 属性创建 InitialContext 对象。
3. **构建 JNDI 名称：** 客户端根据 EJB 的部署信息构建 JNDI 名称。
4. **使用 Context.lookup 查找 EJB：** 使用构建好的 JNDI 名称在 InitialContext 中执行查找操作。
5. **调用远程方法：** 通过返回的 EJB 代理对象调用远程方法。
6. **处理异常和结果：** 处理可能出现的异常情况。

## 4. 碰到的问题及解决方案或思考

碰到的问题： 运行LoginClient.java时出现报错：类文件具有错误的版本 61.0, 应为 52.0 请删除该文件或确保该文件位于正确的类路径子目录中

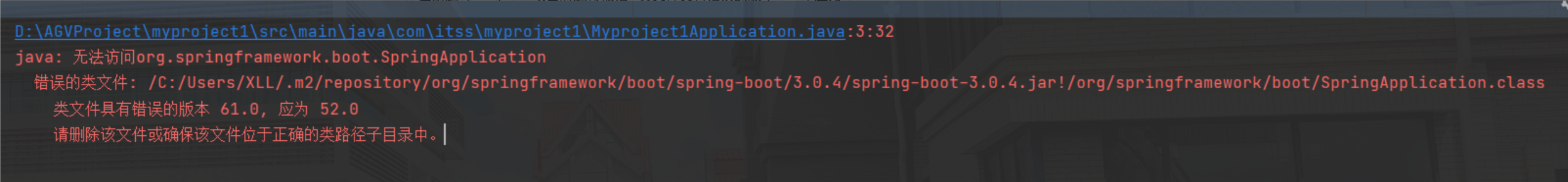


图1.5程序报错

分析原因：打包UserServiceRemote.java类时使用的JDK版本是17，而使用该JAR包时使用的JDK版本是1.8，故出现此报错

解决办法：使用JDK1.8版本打包UserServiceRemote.java类

## 5. 实验收获及总结

在这个实验中，我学到了很多关于 EJB和 JBoss 服务器的内容。EJB 是 JavaEE 服务器端组件模型，而 JBoss 是支持 EJB 的服务器。通过这个实验，我掌握了如何安装和配置 JBoss 服务器，以及如何创建并部署会话 Bean。

在这个过程中，我理解了 EJB 的概念、JNDI 的使用以及如何开发并部署会话 Bean。通过修改登录逻辑和测试不同的用户名密码，我进一步加深了对 EJB 的理解。这个实验让我对企业级 Java 开发有了更深入的了解，并且能够利用 EJB 技术实现分布式系统。