通过Maven集成spring\_springMvc\_hibernate

目录

[1 安装Maven 1](#_Toc534969380)

[2 创建Maven工程 1](#_Toc534969381)

[3 创建源码和配置文件目录 3](#_Toc534969382)

[4 Maven工程转动态Web和配置Tomcat 4](#_Toc534969383)

[5 集成spring-mvc 5](#_Toc534969384)

[5.1 pom.xml配置spring-mvc依赖 5](#_Toc534969385)

[5.2 新建spring-mvc.xml配置文件 7](#_Toc534969386)

[5.3 web.xml中配置spring-mvc 8](#_Toc534969387)

[5.4 创建Controller和jsp测试 8](#_Toc534969388)

[6 集成spring 10](#_Toc534969389)

[6.1 pom.xml配置spring依赖 10](#_Toc534969390)

[6.2 新建spring.xml配置文件 10](#_Toc534969391)

[6.3 web.xml中配置spring 10](#_Toc534969392)

[6.4 创建Service测试 11](#_Toc534969393)

[7 集成hibernate 12](#_Toc534969394)

[7.1 pom.xml配置hibernate依赖 12](#_Toc534969395)

[7.2 spring.xml中整合hibernate 12](#_Toc534969396)

[7.3 创建entity和dao和数据库测试 13](#_Toc534969397)

[8 集成log4j日志系统 15](#_Toc534969398)

[8.1 pom.xml配置log4j依赖 15](#_Toc534969399)

[8.2 新建log4j.properties配置文件 15](#_Toc534969400)

[8.3 web.xml中配置log4j 16](#_Toc534969401)

[8.4 调用log测试 16](#_Toc534969402)

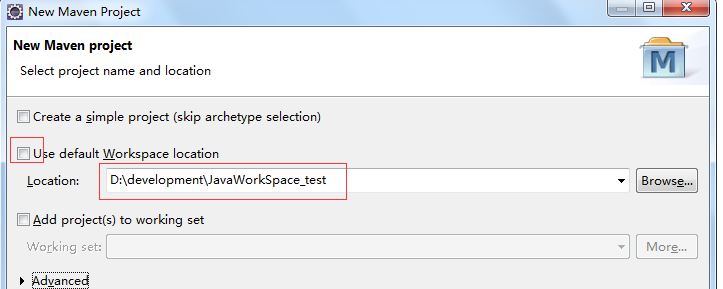
# 1 安装Maven

# 2 创建Maven工程

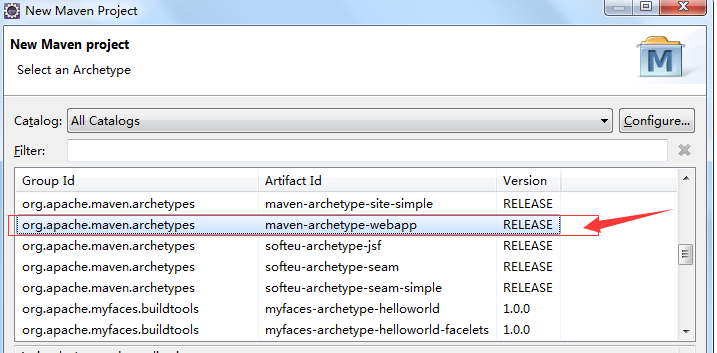
Eclipse > file > new > project > MavenProject



点击 Browser选择项目工作空间目录

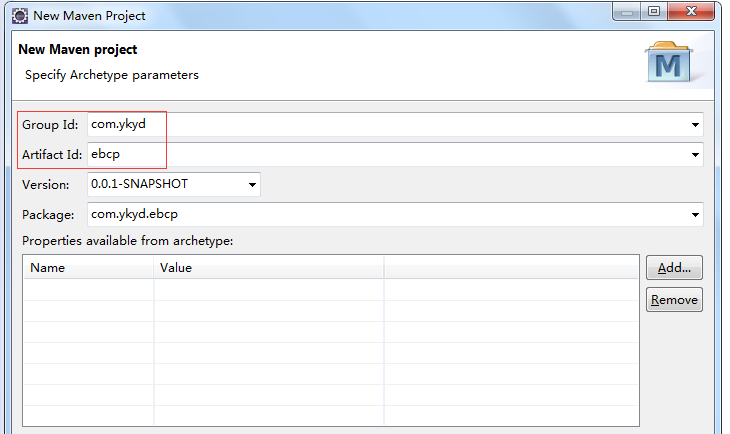


Archetype选择webapp



Group Id ：com.公司名称，例如：com.ykyd

Aritifact Id：项目名称，例如：ebcp 电商云平台

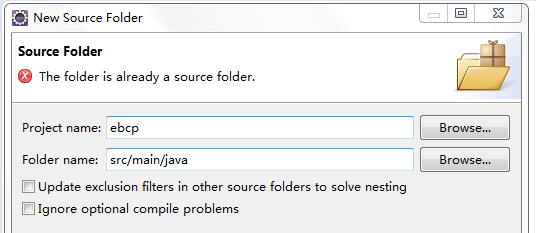


# 3 创建源码和配置文件目录

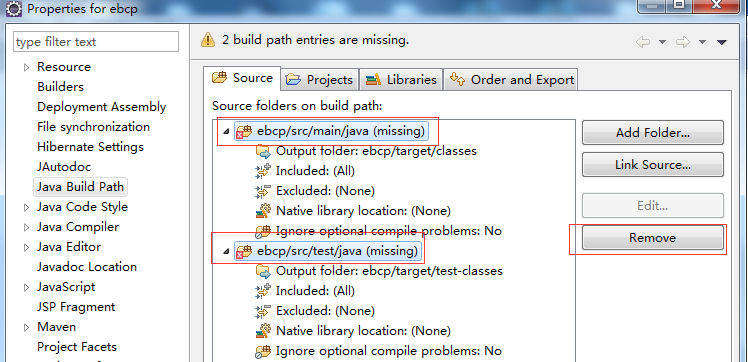
在创建好的工程根目录下面创建src/main/java目录，用于存放java源代码；

创建src/main/resource目录，用于存放各种配置文件

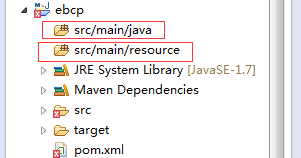
项目右键 > new > Source Floder



注意：此时会报错目录已存在，项目右键 > Properties > Java Build Path > Source ，将报红叉的全部移除，再重新创建源码目录和配置文件目录

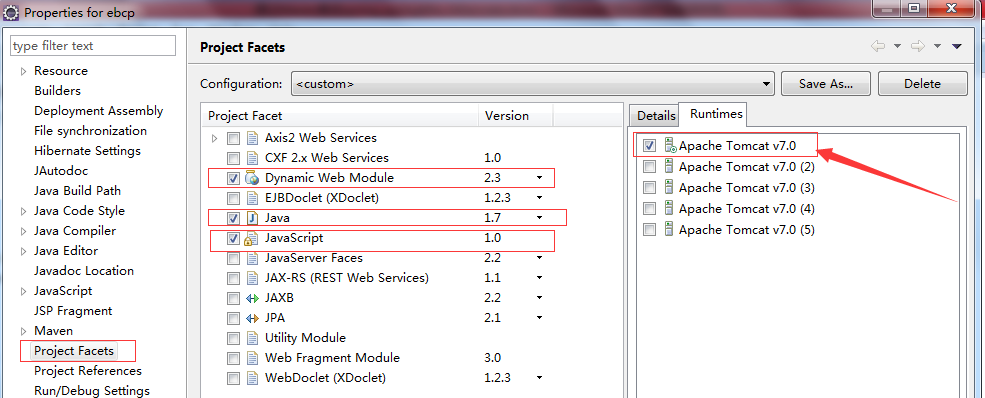


创建好的目录如下图

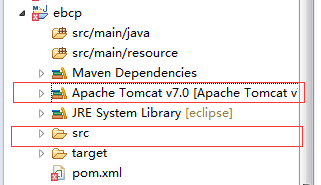


# 4 Maven工程转动态Web和配置Tomcat

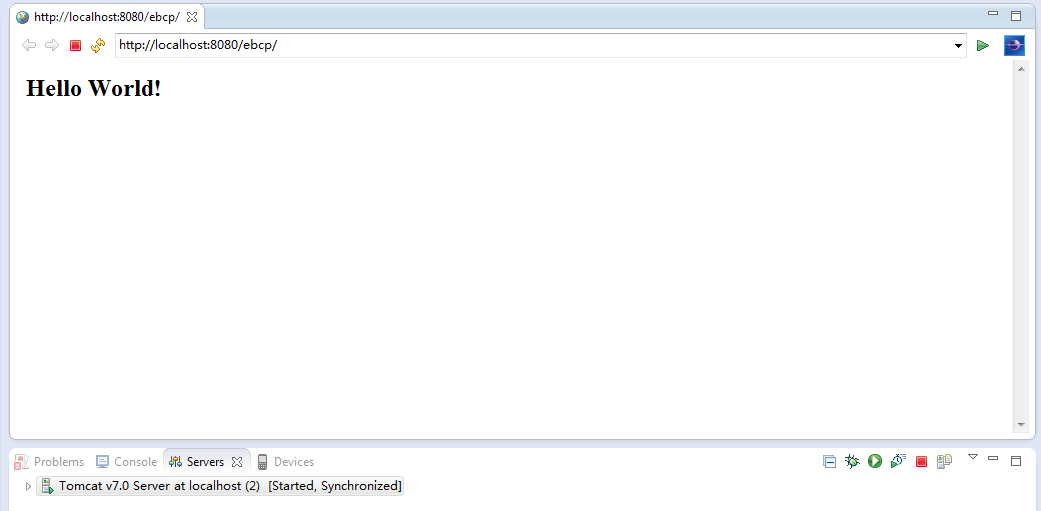
项目右键 > Properties > Project Facets



配置好了的项目目录如下图，多了Apache Tomcat ，并且src目录上的报错也消失了



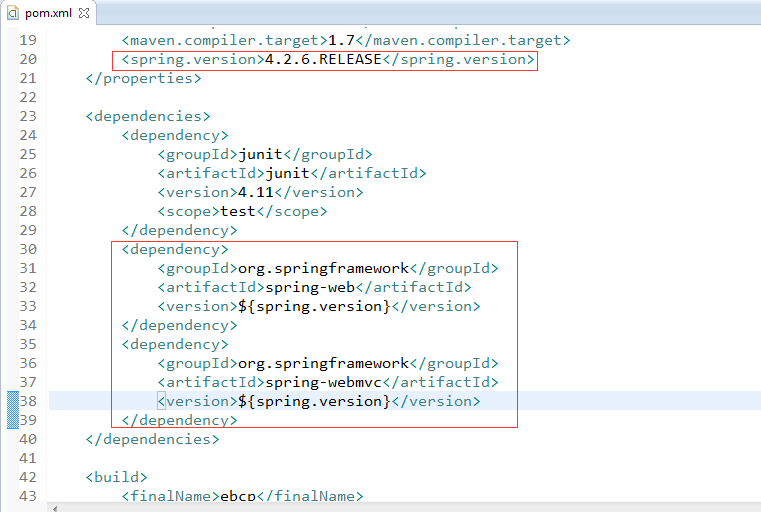
此时项目也可以运行了



# 5 集成spring-mvc

## 5.1 pom.xml配置spring-mvc依赖

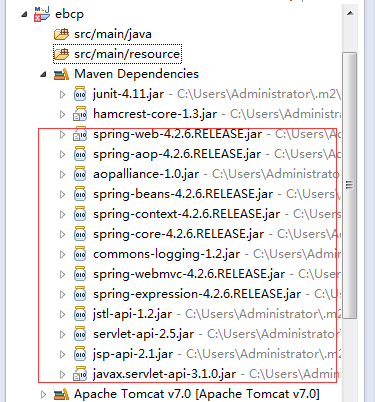
在pom.xml文件的添加依赖spring-web 和spring-mvc



再引入jstl-api 和 javax.servlet-api

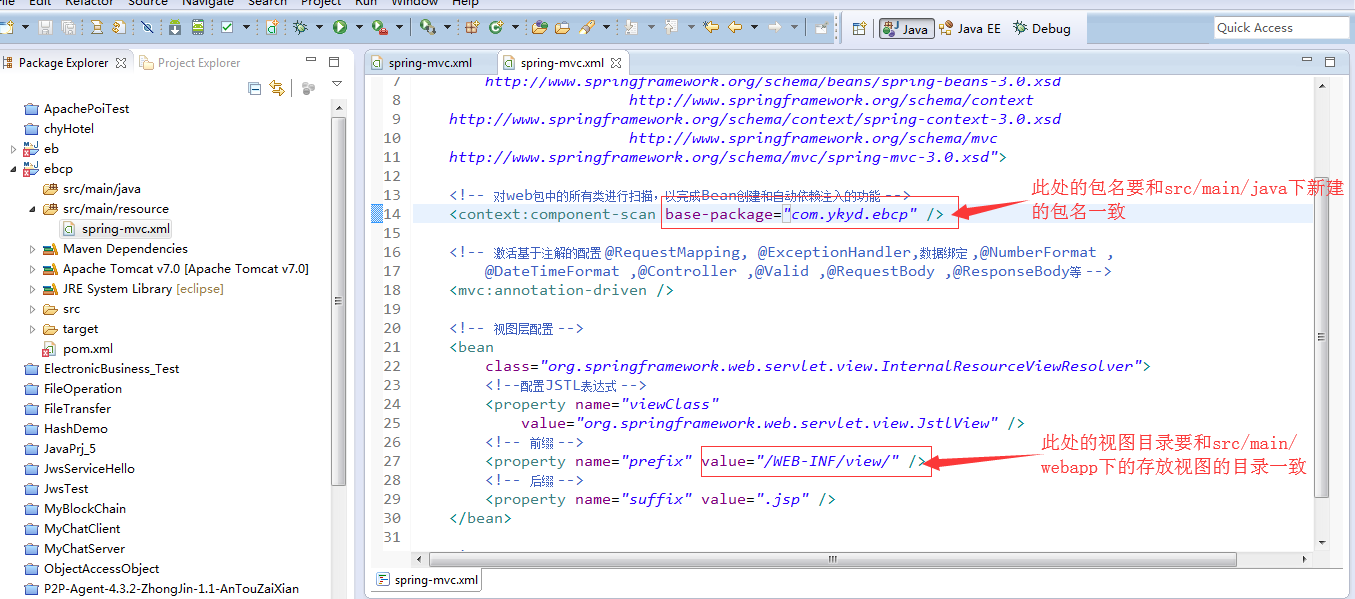


Ctrl +S 保存后，系统会自动去下载依赖包



## 5.2 新建spring-mvc.xml配置文件

在src/main/resources中新建一个spring-mvc.xml的文件



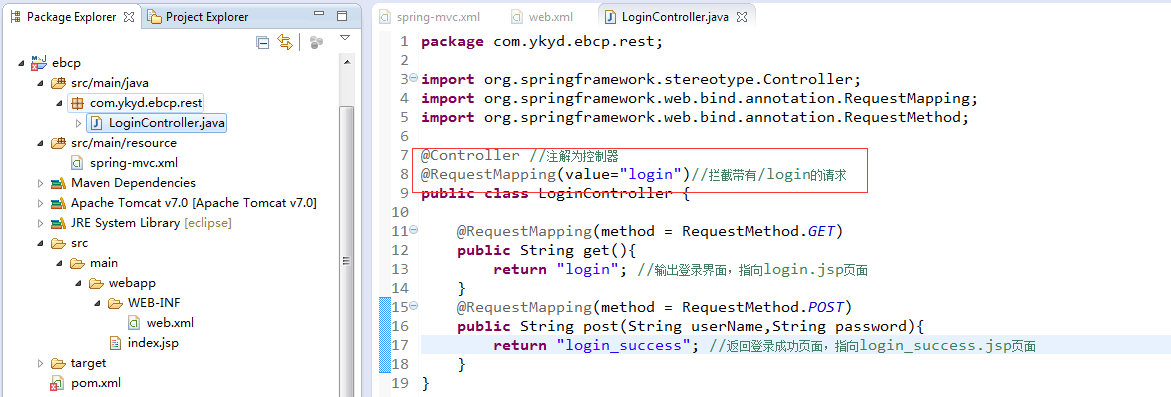
## 5.3 web.xml中配置spring-mvc

在web.xml文件中配置spring-mvc相关

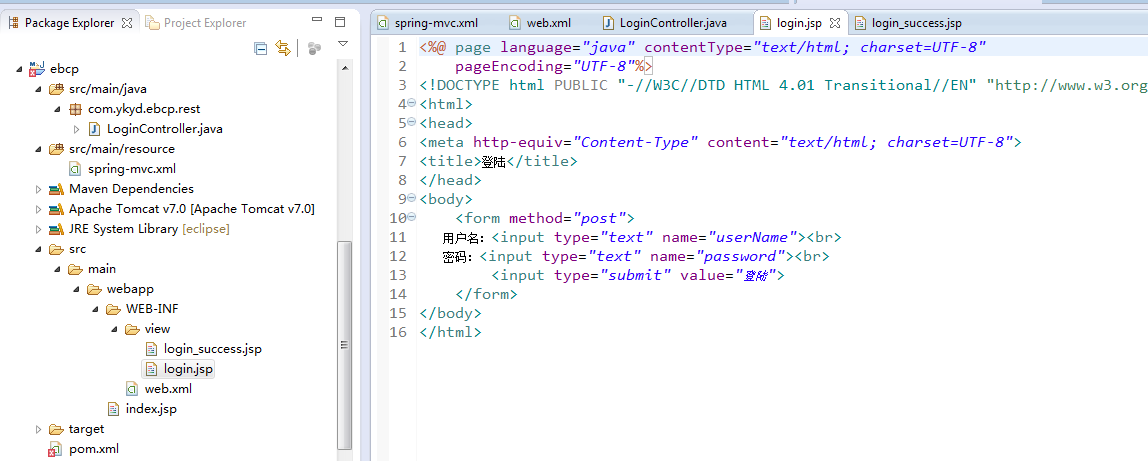


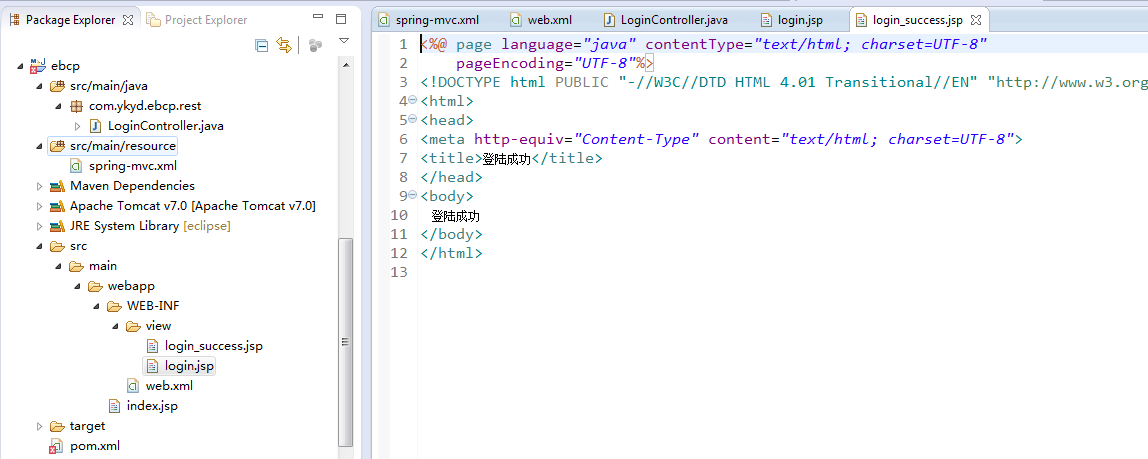
## 5.4 创建Controller和jsp测试

在src/main/java目录下新建包名，如com.ykyd.ebcp.rest，在这个包里新建一个LoginControllrer.java作为控制器

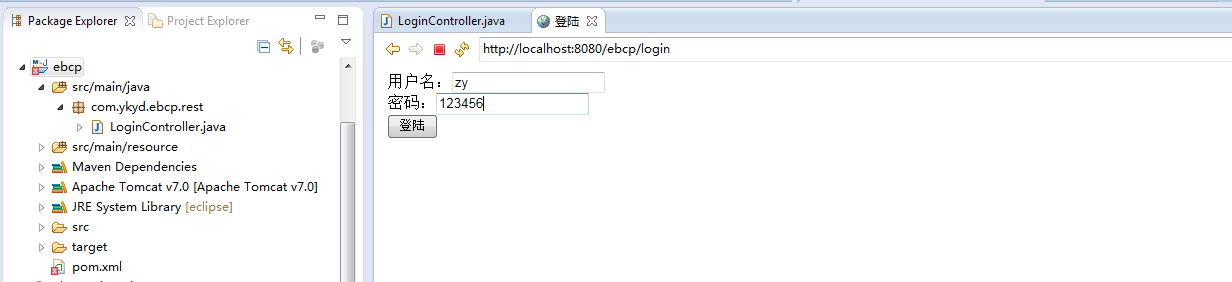


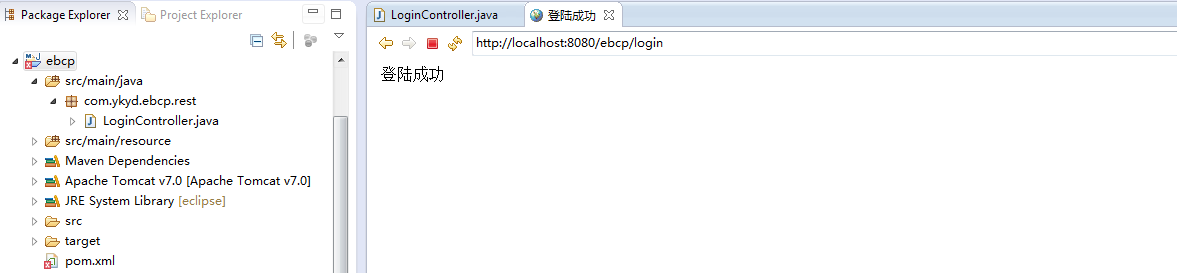
在src/main/webapp/WEB-INF目录下新建view目录，然后分别创建login.jsp和login\_success.jsp文件





运行项目测试成功

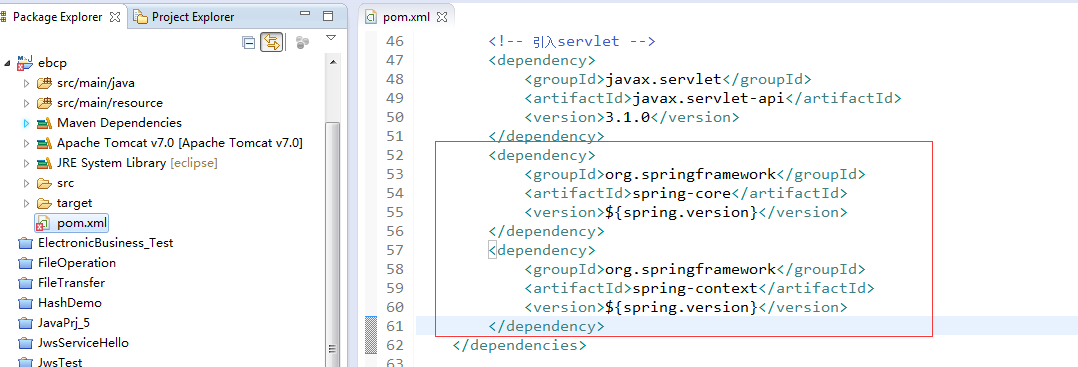




# 6 集成spring

## 6.1 pom.xml配置spring依赖

在pom.xml文件的添加依赖spring-core 和spring-context



Ctrl +S 保存后，系统会自动去下载依赖包

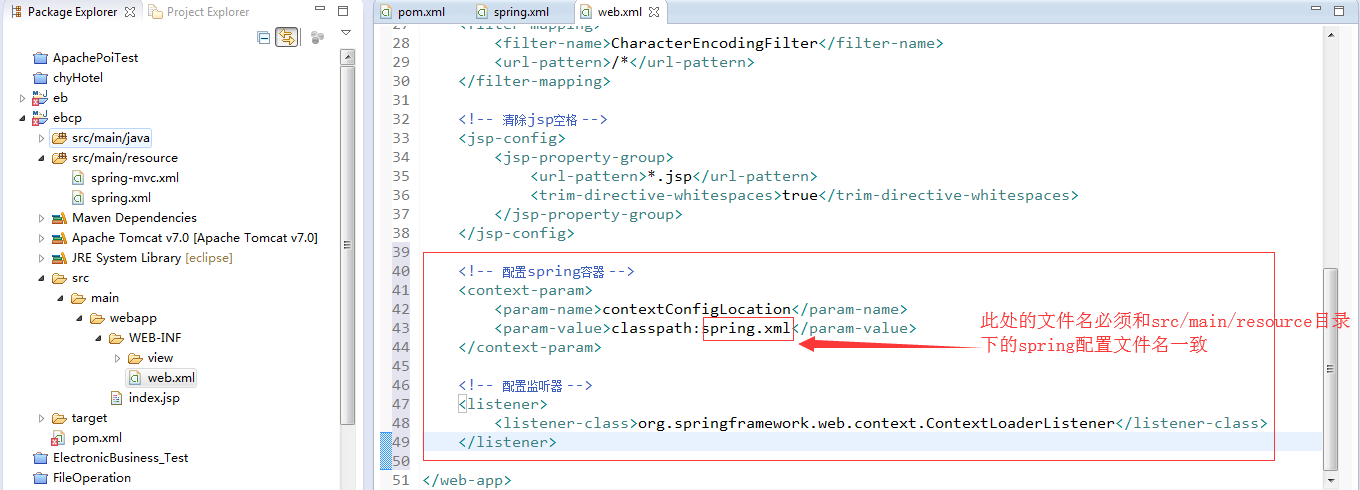
## 6.2 新建spring.xml配置文件

在src/main/resource目录下新建spring.xml配置文件



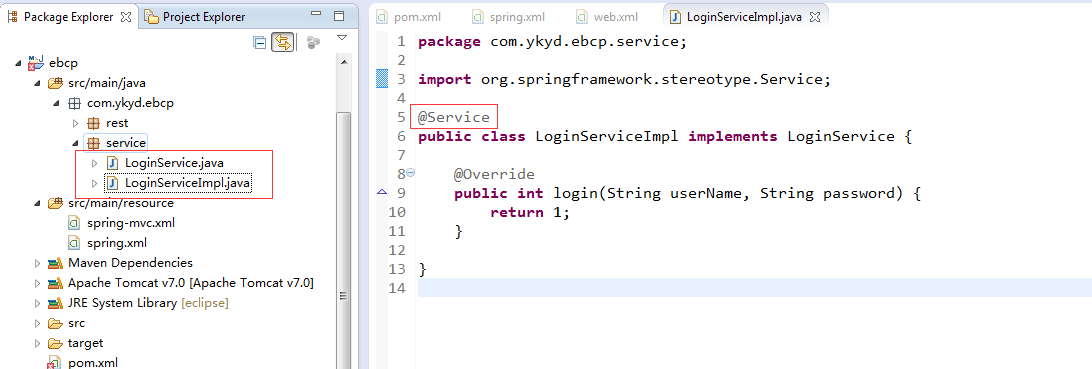
## 6.3 web.xml中配置spring

在web.xml文件中配置spring相关

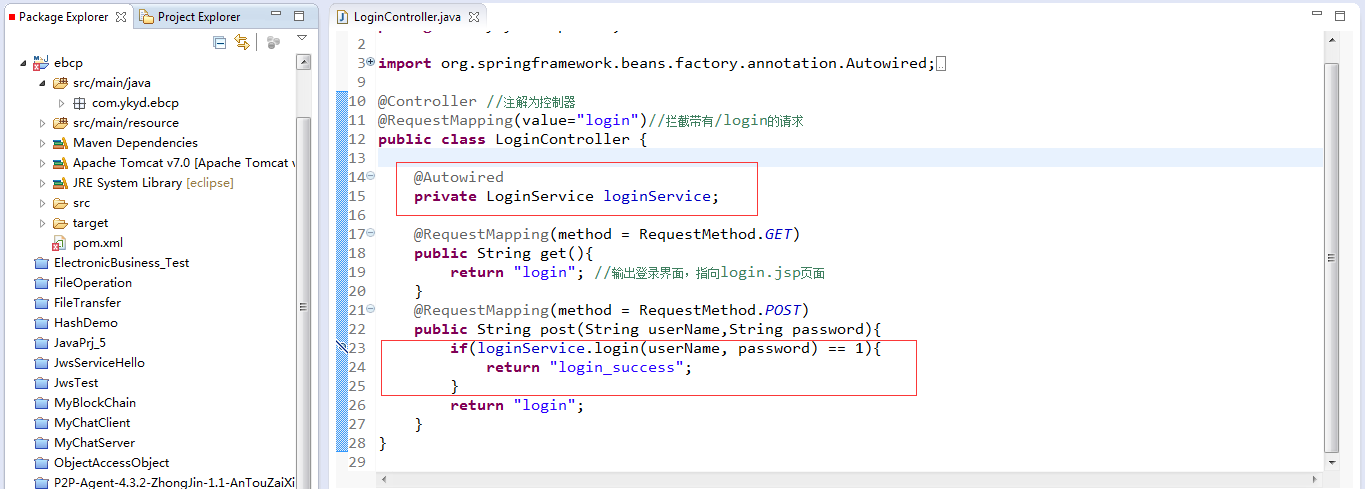


## 6.4 创建Service测试

在src/main/java目录下新建包名：com.ykyd.ebcp.service，并在此包中创建service接口和实现类



在LoginController中调用Service

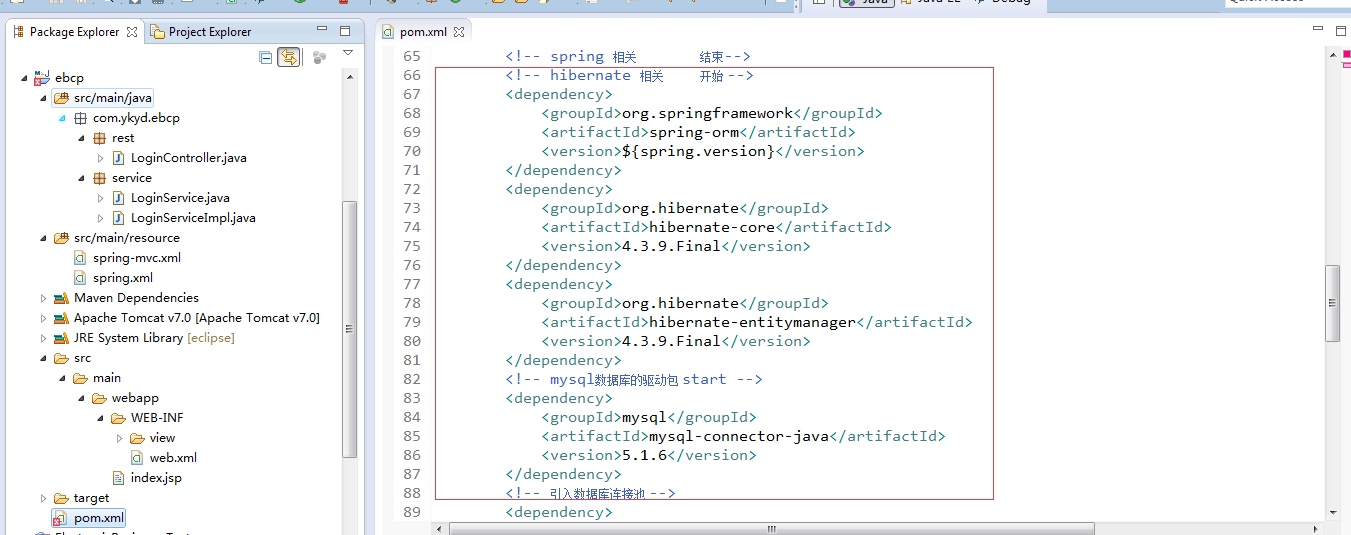


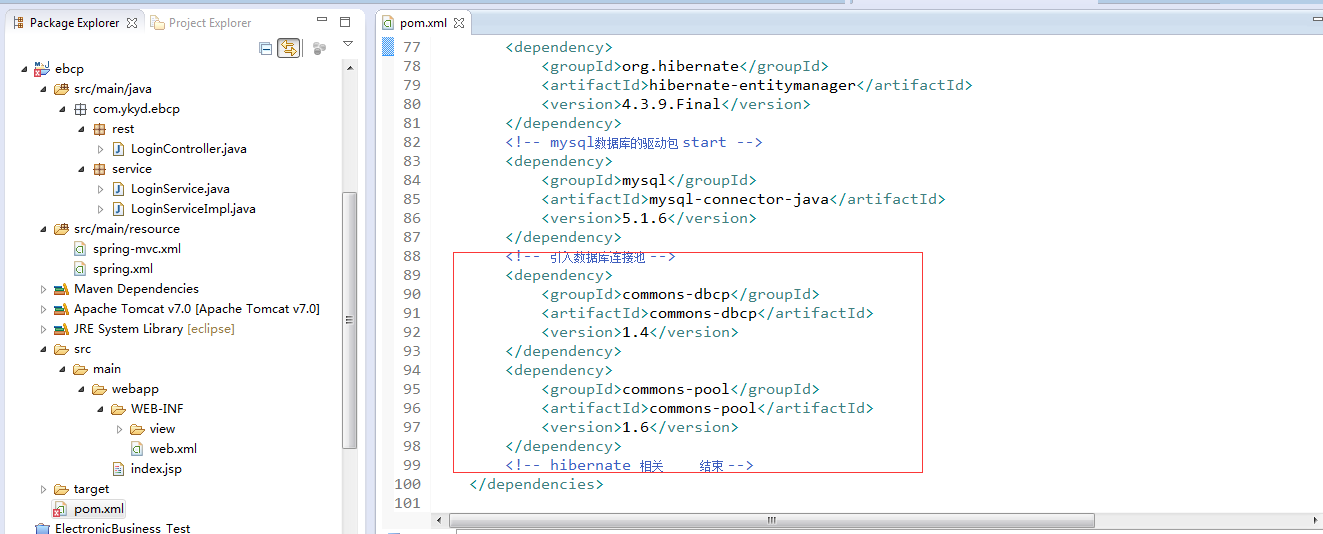
运行项目，测试成功

# 7 集成hibernate

## 7.1 pom.xml配置hibernate依赖

在pom.xml文件中添加spring-orm、hibernate-core、hibernate-entitymanager、mql数据库驱动和数据库连接池

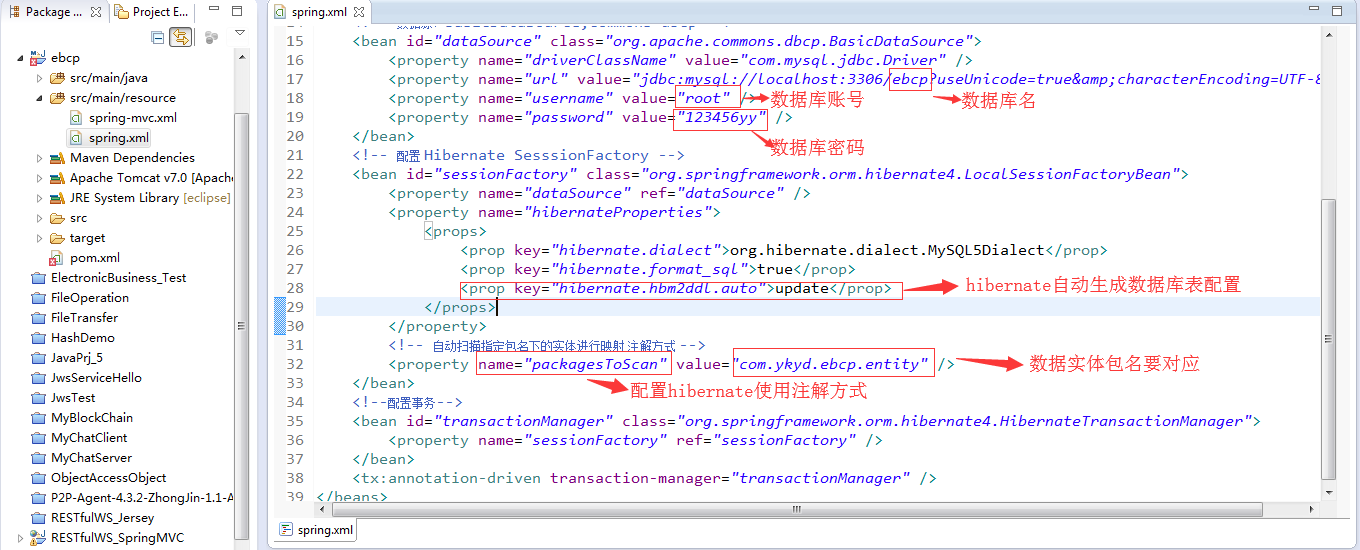




Ctrl + S 保存后系统会自动下载依赖包

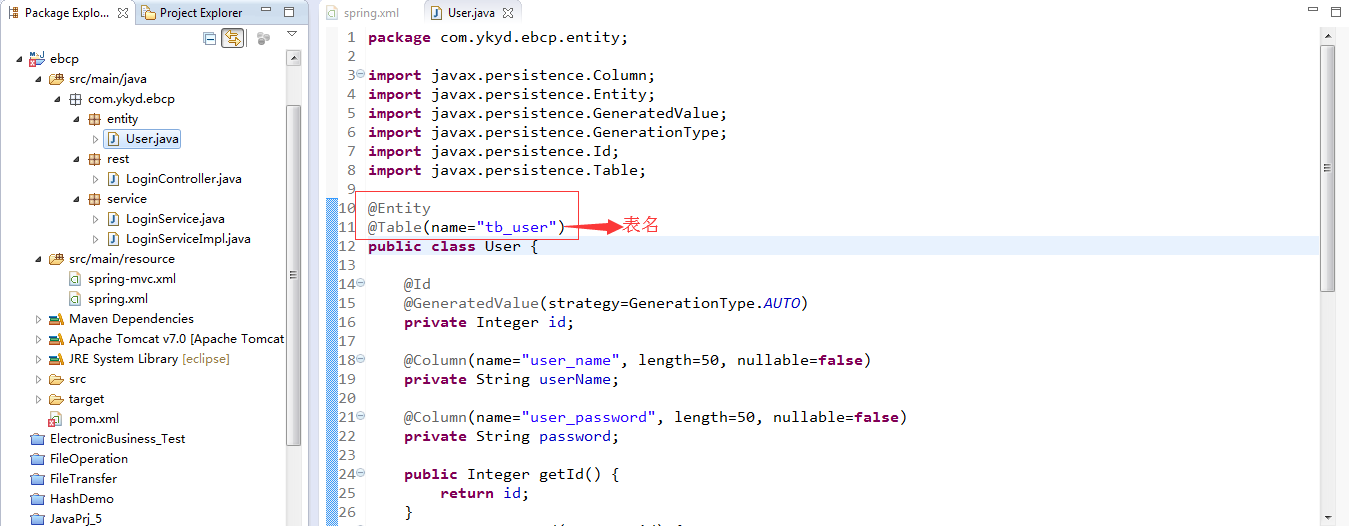
## 7.2 spring.xml中整合hibernate

在spring.xml文件中配置数据库连接相关，配置hibernate相关，其中包括是否逆向生成表结构，是否使用注解方式，事务等

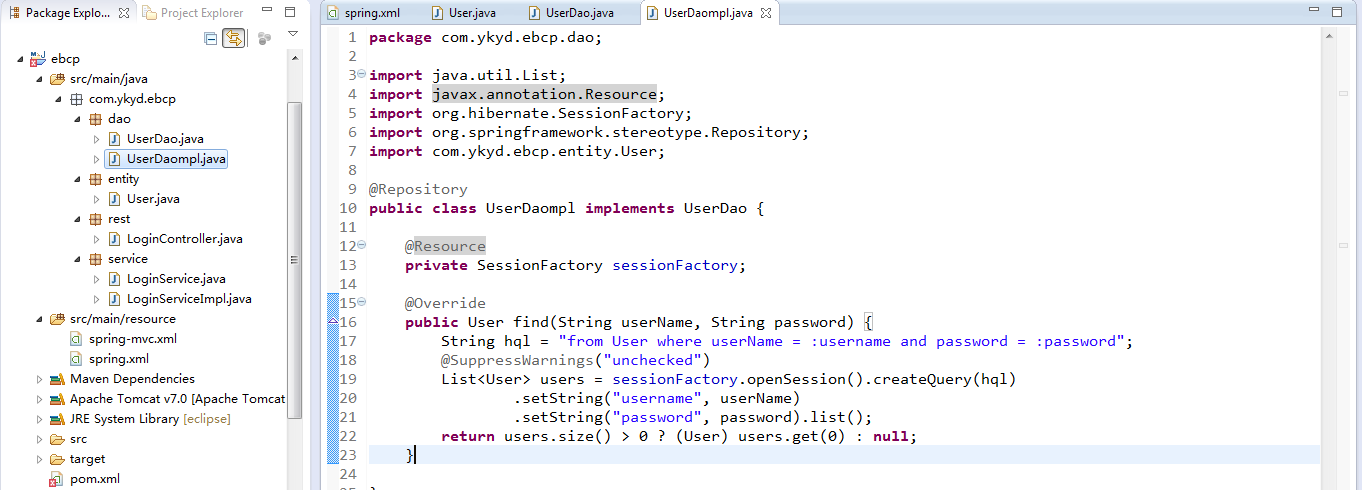


## 7.3 创建entity和dao和数据库测试

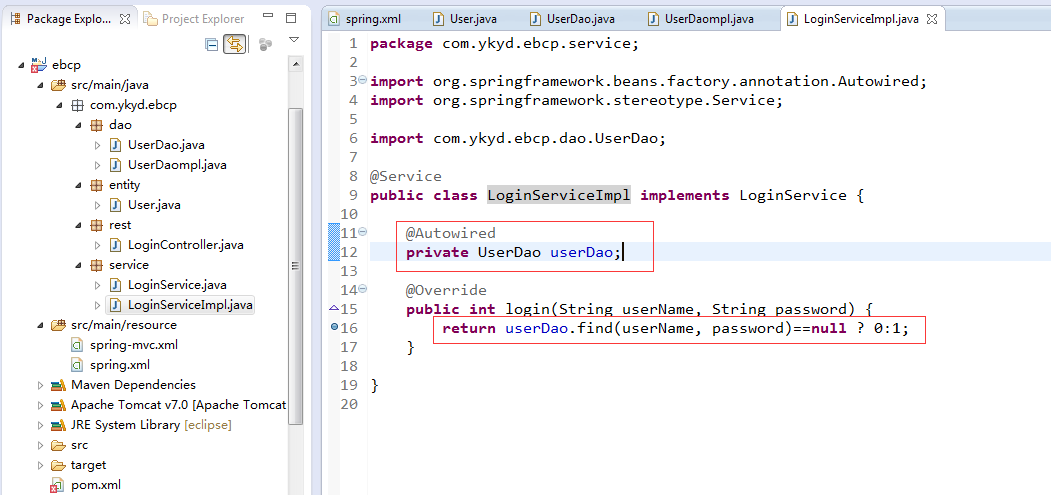
在src/main/java目录下新建com.ykyd.ebcp.entity包，在此包中新建User实体



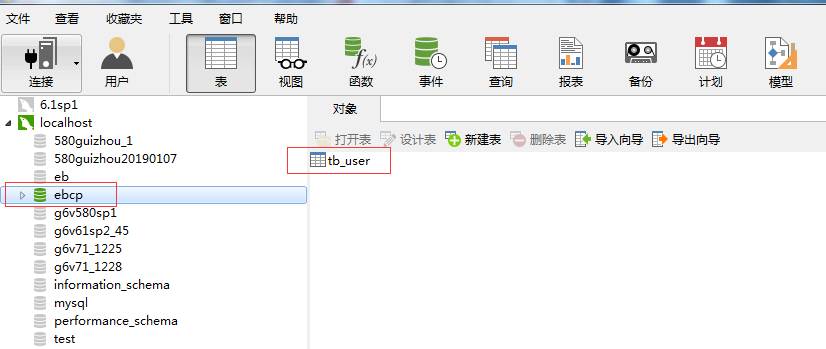
在src/main/java目录下新建com.ykyd.ebcp.dao包，在此包中新建UserDao接口和UserDaoImpl实现类



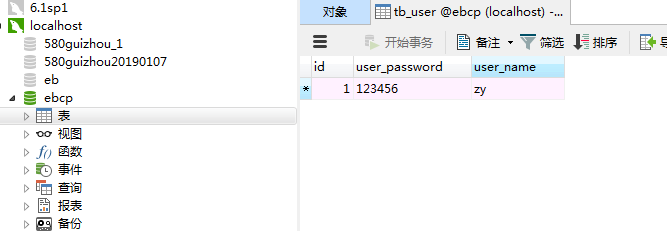
在LoginService中调用UserDao



在数据库中新建一个空的数据库，名称要和spring.xml中配置的一致，本例中取名为ebcp,然后运行项目,数据库中生成了tb\_user表，表示集成成功



手动在tb\_user中添加id，user\_name，user\_password，然后登陆测试，登陆成功



至此hibernate集成已全部完成。

# 8 集成log4j日志系统

## 8.1 pom.xml配置log4j依赖

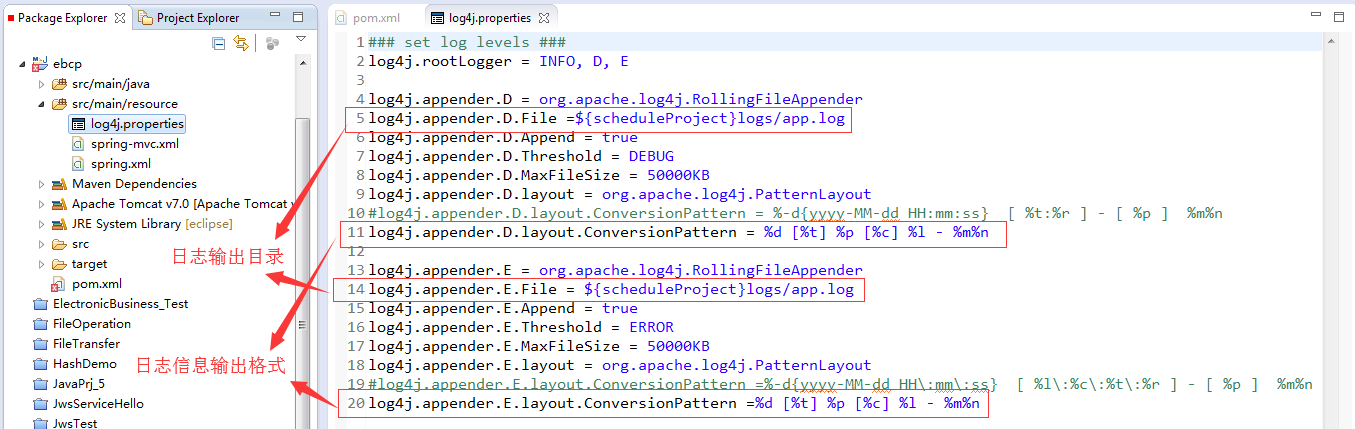
在pom.xml文件中添加log4j依赖



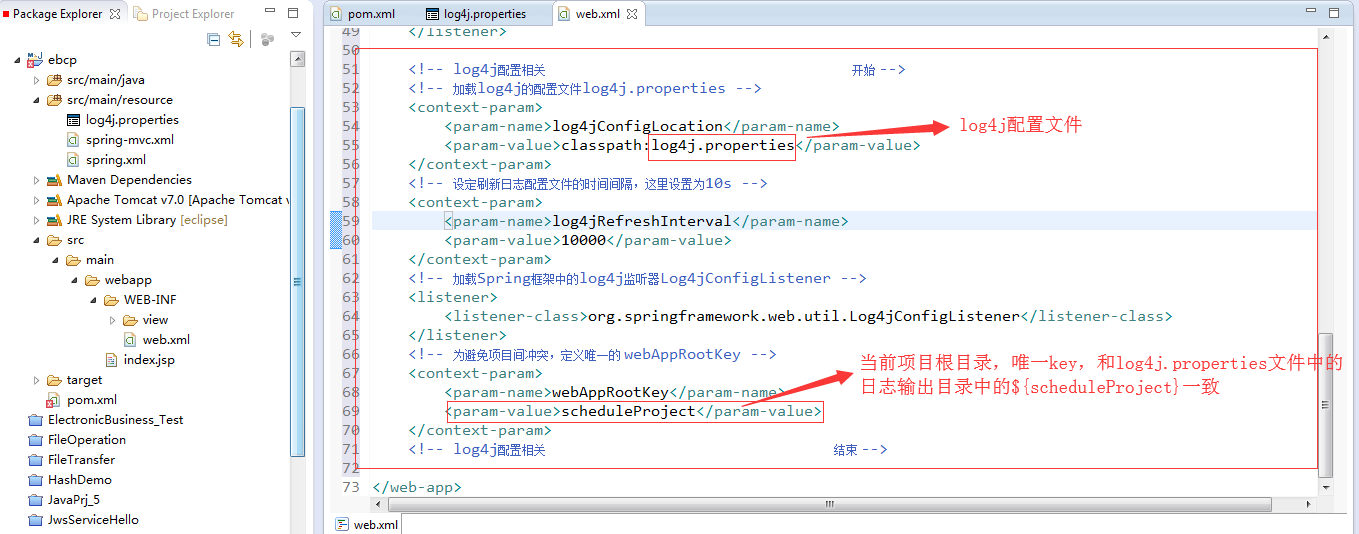
Ctrl + S 保存后系统会自动下载依赖

## 8.2 新建log4j.properties配置文件

在src/main/resource目录下新建log4j.properties配置文件

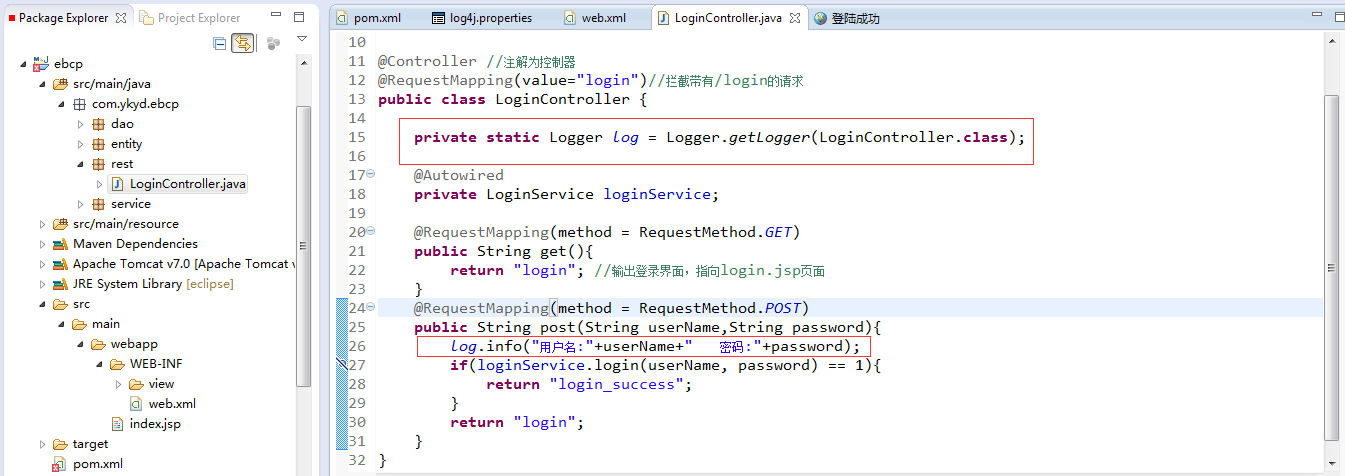


## 8.3 web.xml中配置log4j



## 8.4 调用log测试

在UserDaoImpl中调用log输出日志信息



运行项目，测试登录，然后在项目根目录下查看日志已成功输出

