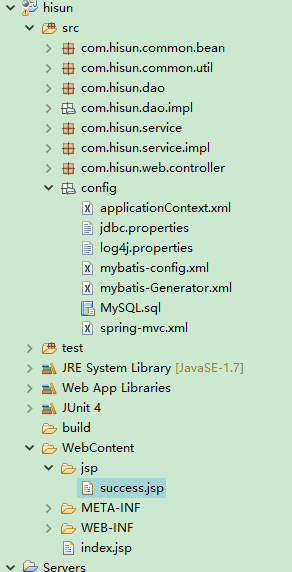
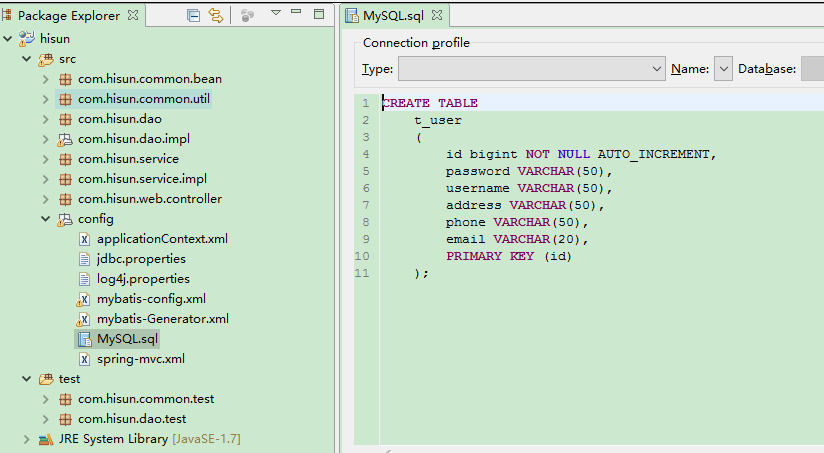
欢迎使用

1. 框架组成
2. SpringMVC + Spring + Mybitas
3. 目录如下：

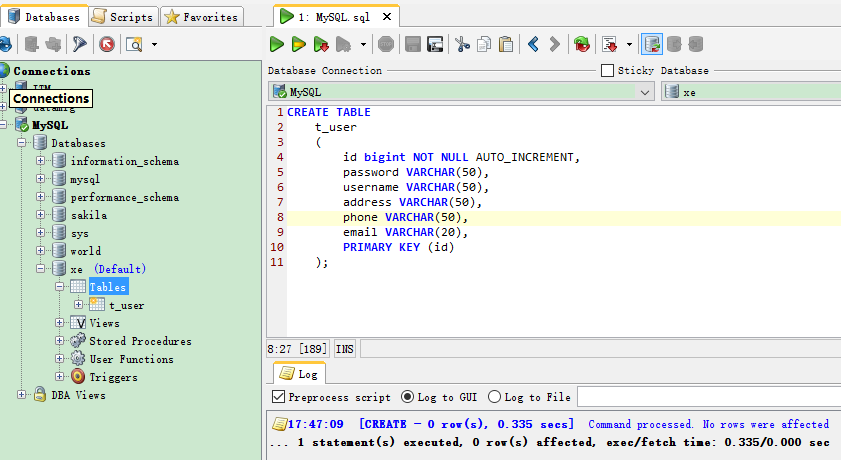


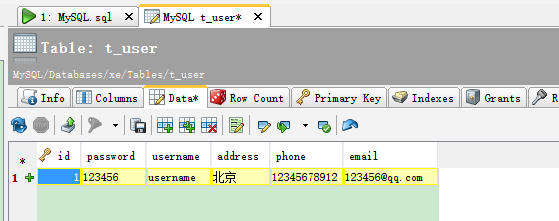
1. 使用步骤

1.打开配置文件config目录下的MySQL.sql文件，本demo以MySQL数据库为例，在文件中编写的建表SQL语句，具体如图下：

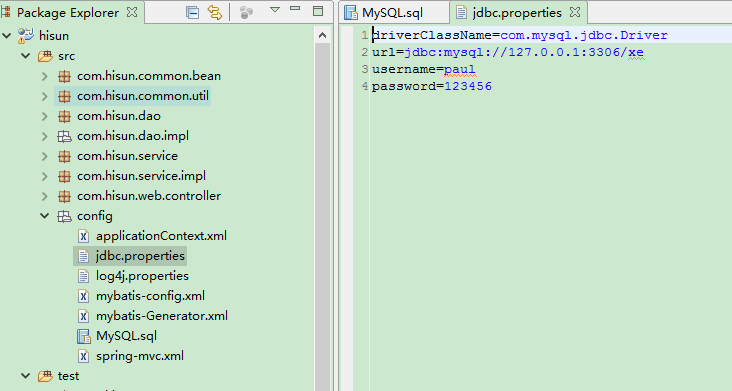


1. 把编写好了的SQL语句，复制并粘贴到数据库客户端，执行SQL语句，随便在建好的表中添加一条数据，具体如图下：

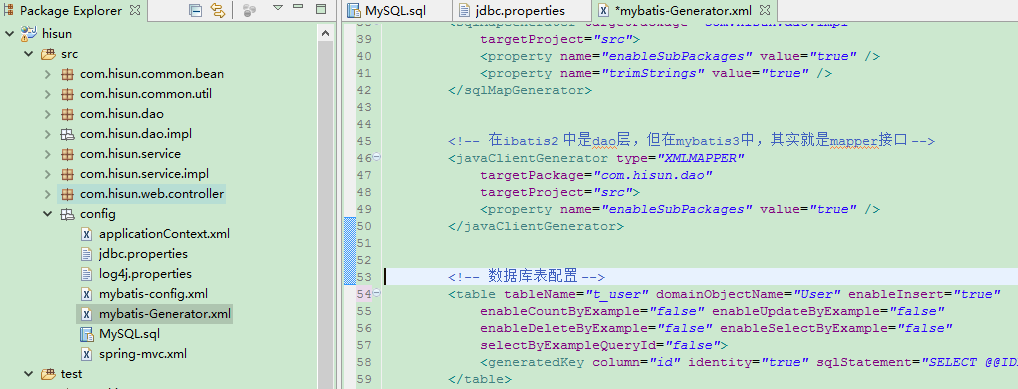




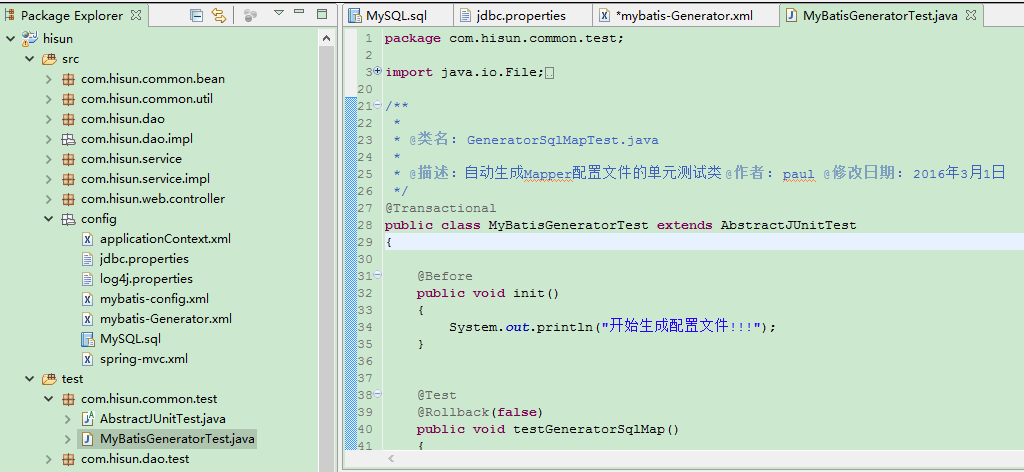
4. 打开配置文件config目录下的jdbc.properties文件，在这里编写MySQL数据库的“数据库驱动”，“路径”，“用户名”和“密码”，具体如图下：



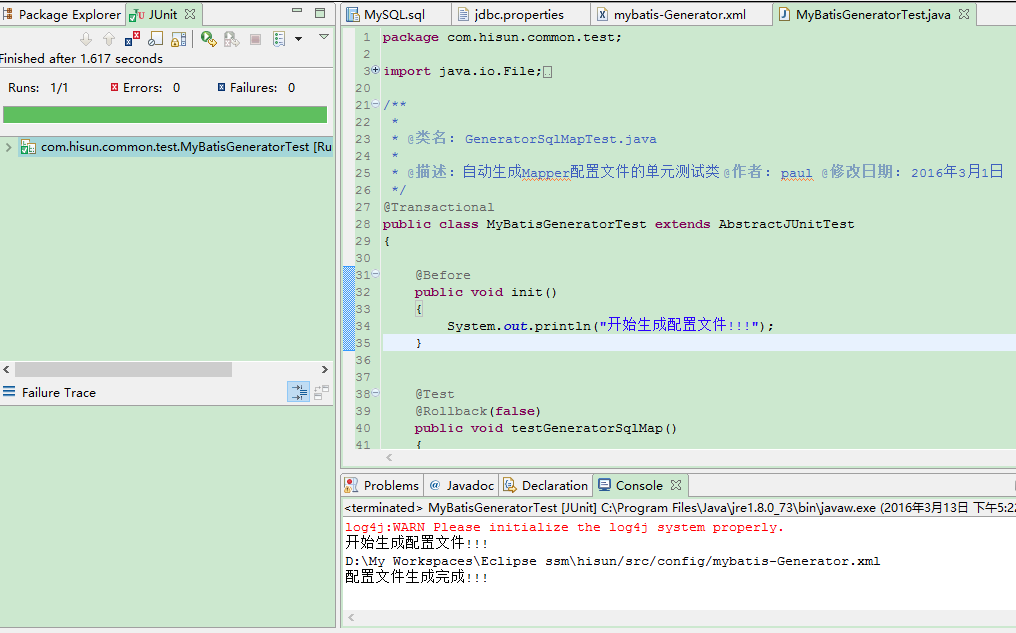
5. 打开配置文件config目录下的mybatis-Generator.xml文件，找到“数据库表配置”位置，编写“数据库表名”及要生成的映射“类名”，具体如图下：



6. 打开单元测试test目录下的MyBatisGeneratorTest.java文件，该文件不需要任何的修改，具体如图下：



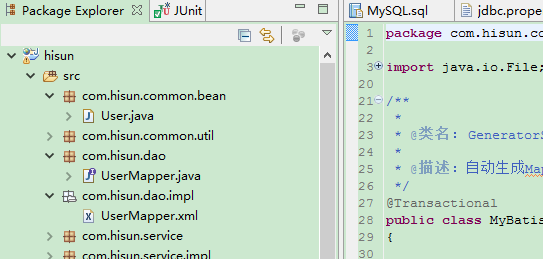
7.选定MyBatisGeneratorTest.java文件，使用Junit Test运行该文件，具体如图下：



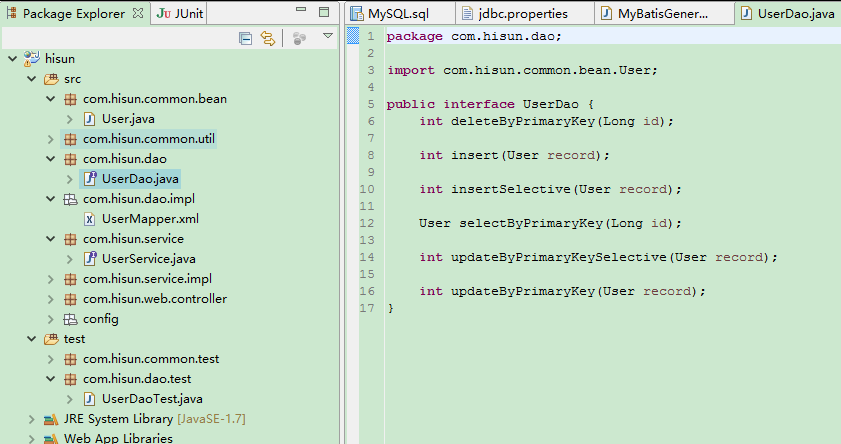
8.根据上面运行结果，说明程序运行成功，如果运行出错，错误原因可能为：

1.数据库连接配置出错 2. MyBatisGeneratorTest.java路径出错

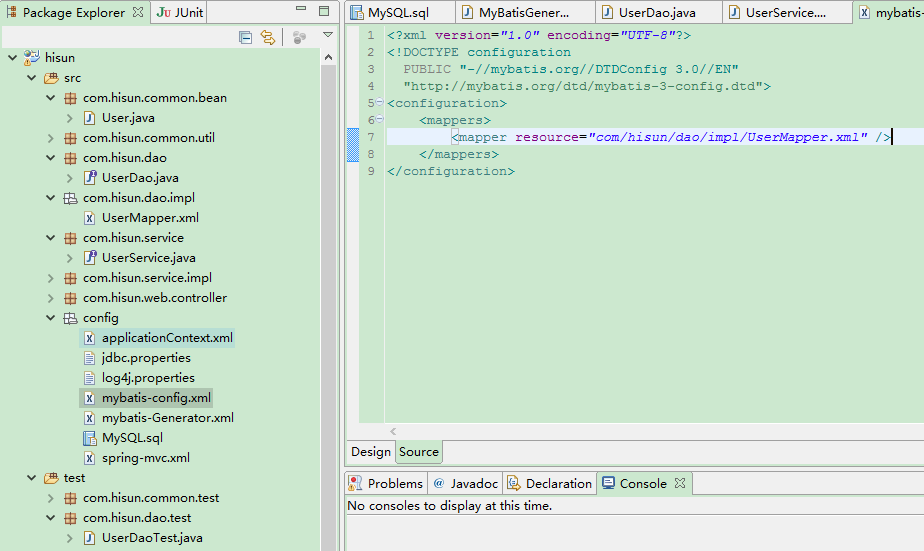
3.表名出错 请认真检查上面三个事项，程序运行成功后，将会产生三个配置文件，分别为bean层User.java文件，dao层UserMapper.java文件，以及UserMapper.xml文件，具体如图下：



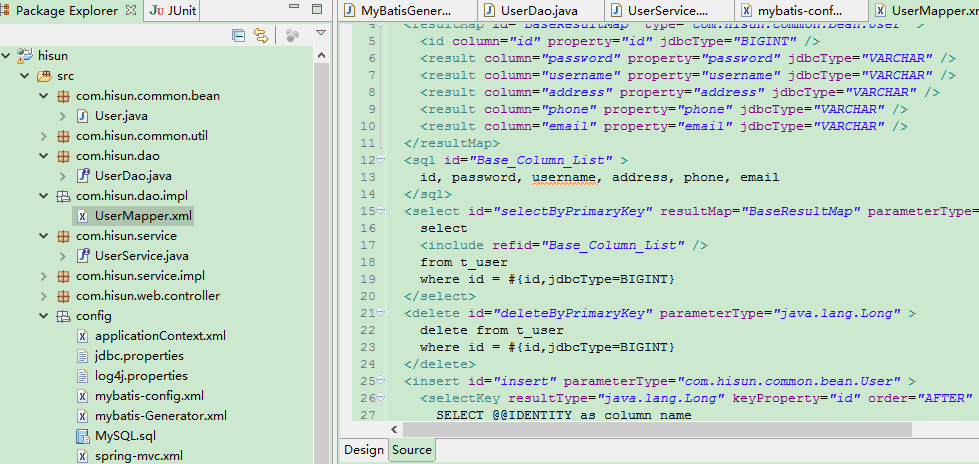
9. 为了好理解我们把dao层UserMapper.java，改名为UserDao.java,这样就一目了然它是属于dao层的文件，具体如图下：



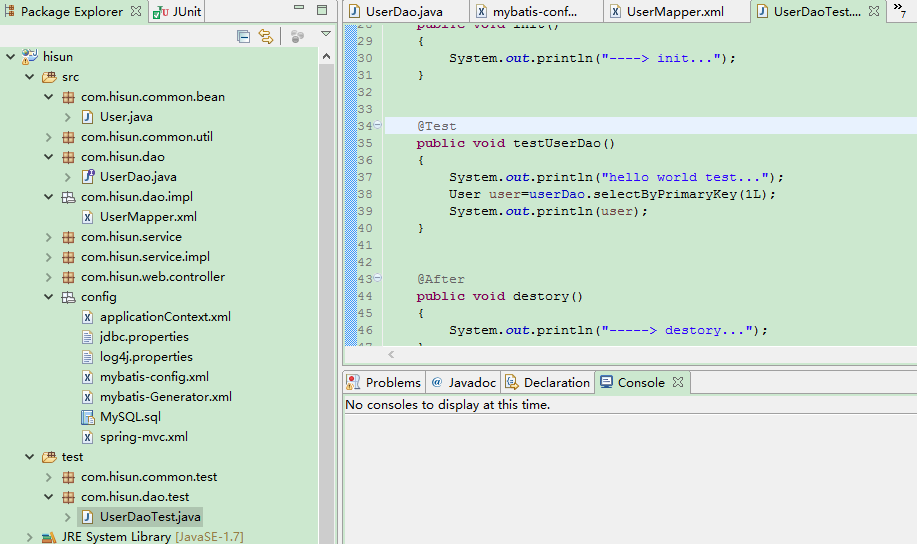
10. 打开配置文件config目录下的mybatis-config.xml文件，添加刚才生成的UserMapper.xml配置文件，具体如图下：



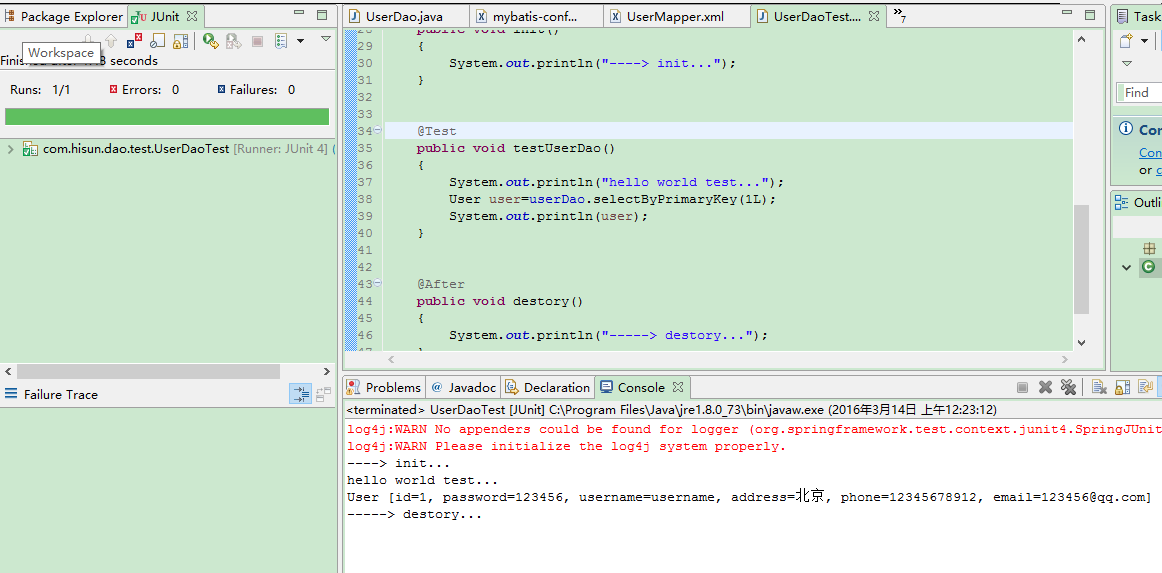
11.下面是UserMapper.xml自动生成的一次数据库操作SQL语句，别名和dao层一样，我们就拿其中一个来单元测试一下，具体如图下：



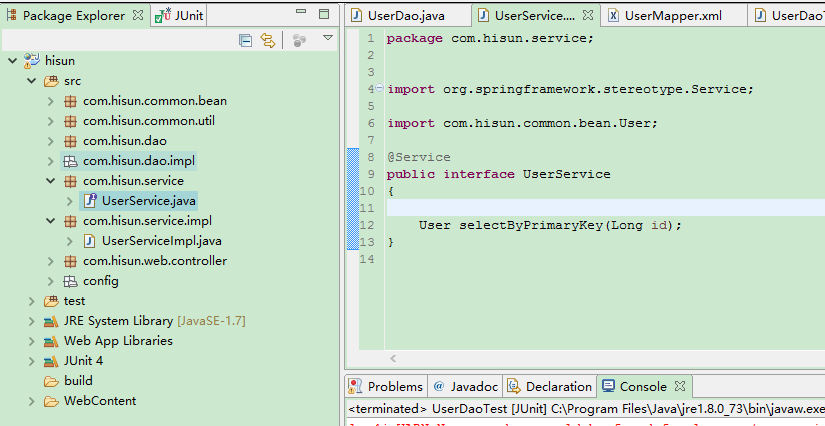
12. 打开test目录下的UserDaoTest.java文件，单元测试dao层其中的”selectByPrimaryKey()”接口，编写UserDaoTest.java文件，具体如图下：



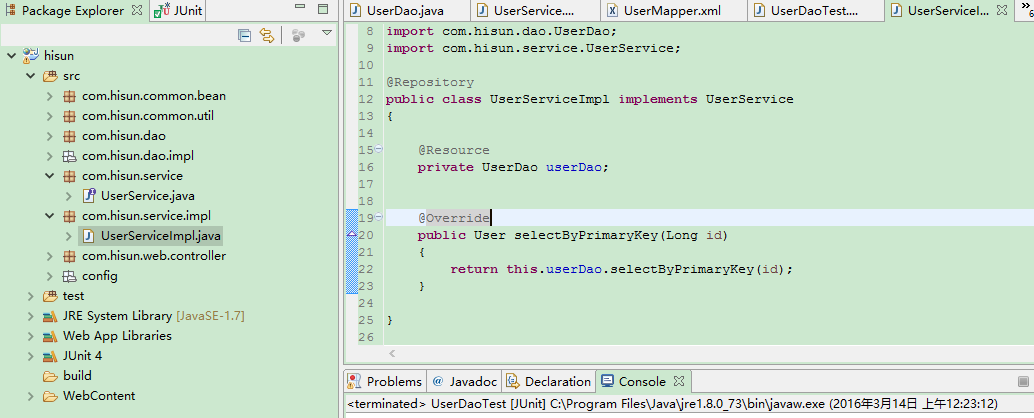
13. 选定UserDaoTest.java文件，使用Junit Test运行该文件，具体如图下：



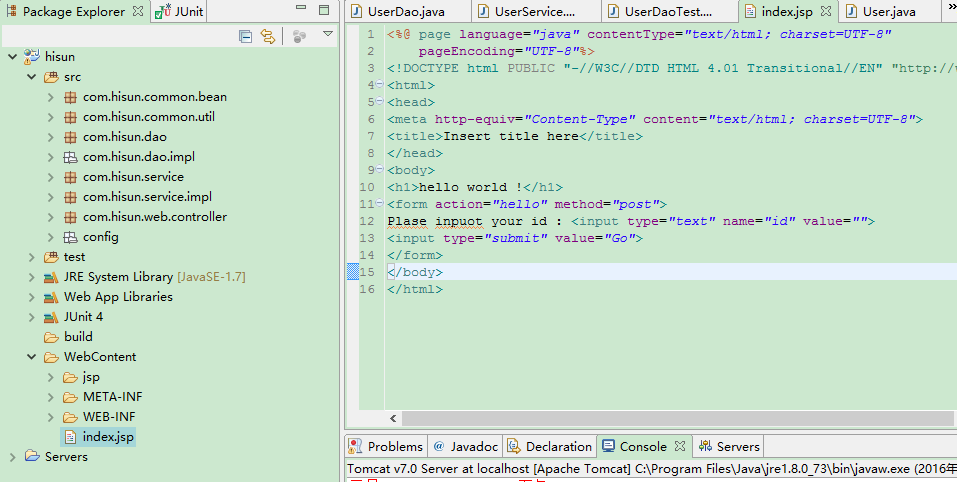
14.到此数据层已编写完成，下面我们来编写service层，编写接口，具体如图下：



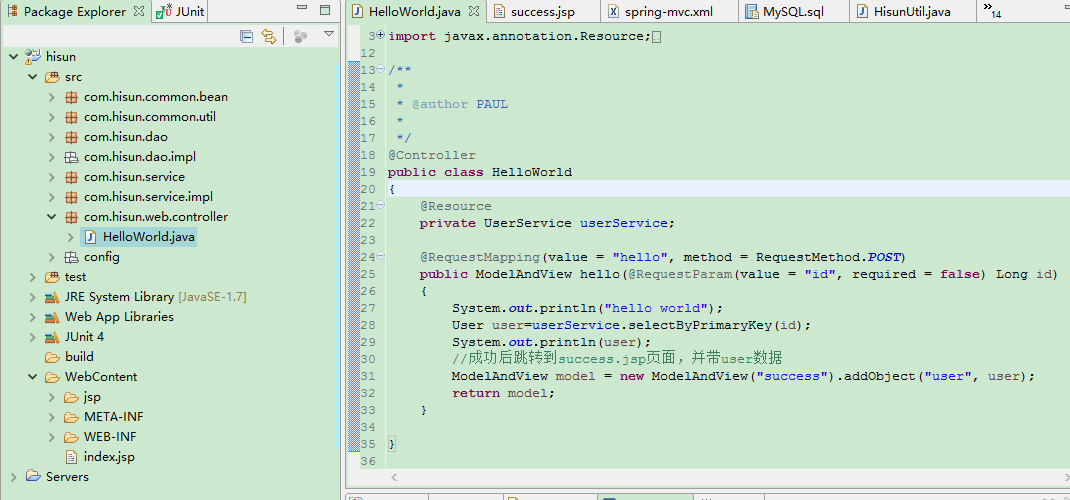
15. service层，编写实现类，具体如图下：



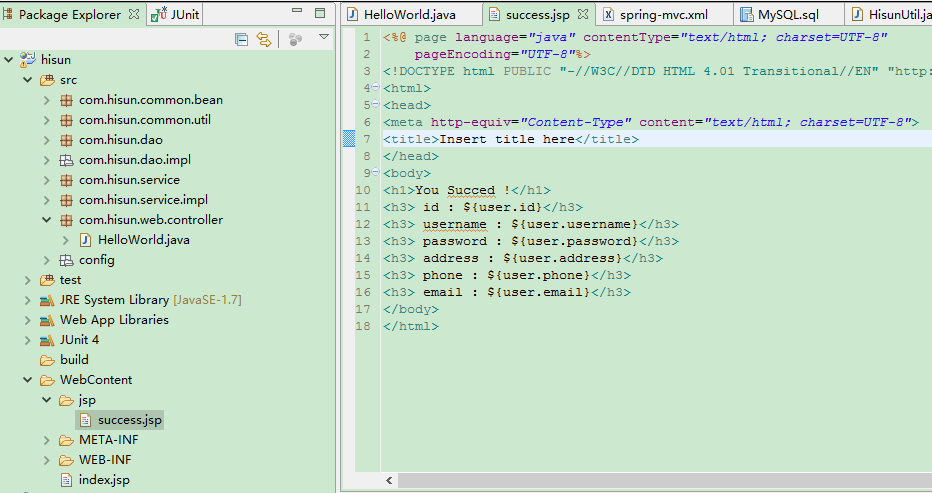
16.这空写没意思，我们就写个经典程序“hello world !“吧，哈哈，那先编写个首页index.jsp，具体如图下：



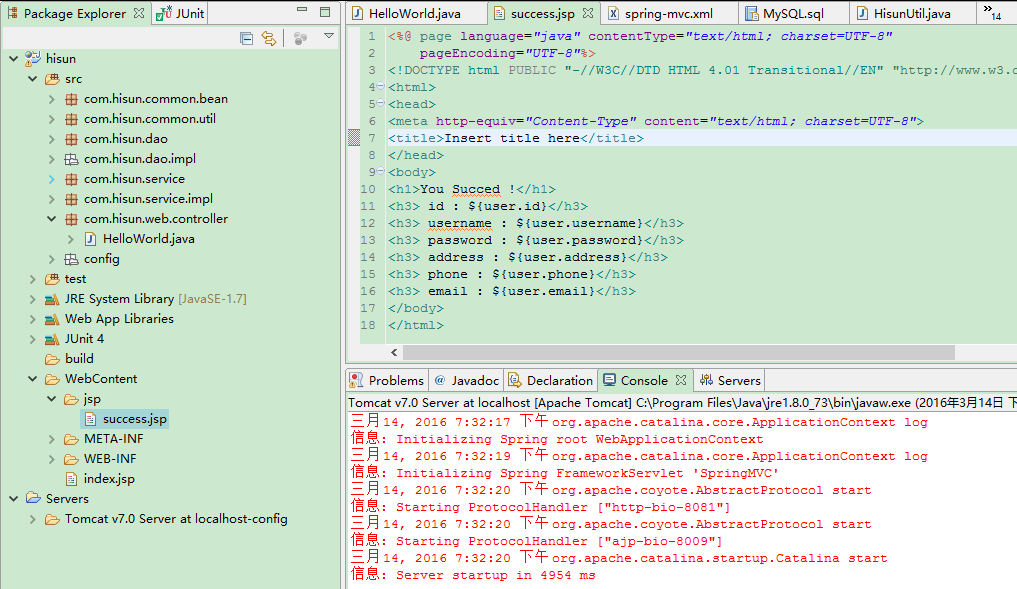
17.再来编写controller层程序，具体如图下：

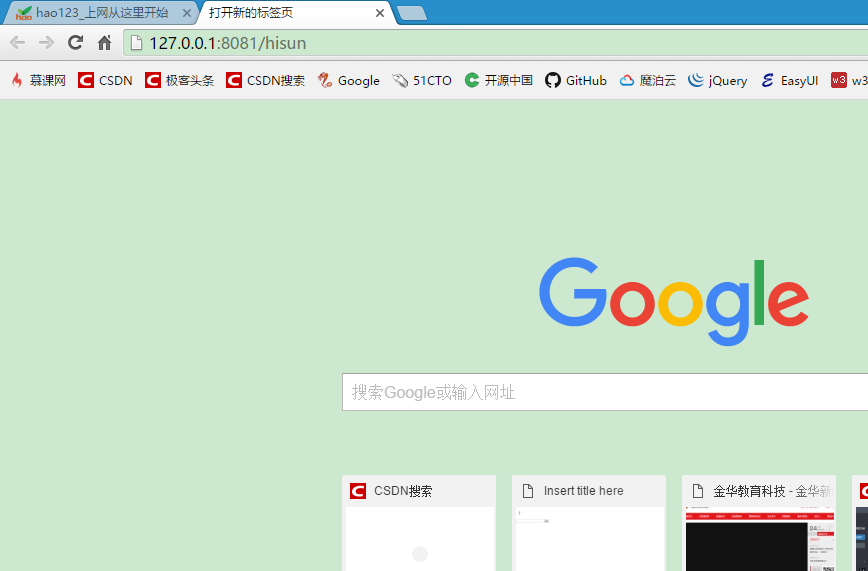


19.接着再编写success.jsp页面，具体如图下：

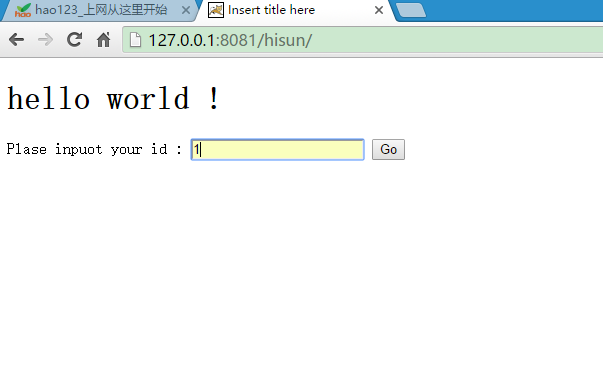


20.最后配置Tomcat,运行Server，没有报错，项目运行成功，具体如图下：

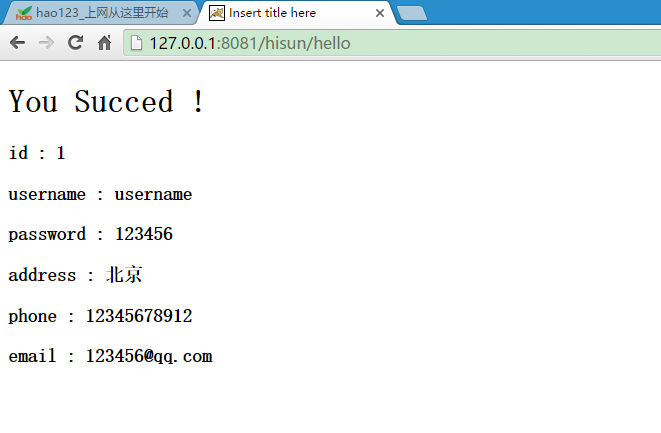


21. 打开浏览器输入网址：127.0.0.1:8081/hisun，8081根据你的Tomcat配置而定，具体如图下：

22.进入首页index.jsp, 并输入当初建时，插入的一条数据的id,具体如图下：



23. 再点击Go按钮，成功将调到success.jsp页面，具体如图下：



24.好了由于本人经历有限就写到这里，如有哪里有误，多多包涵。谢谢！!