Spring

**SSM三大框架整合详细教程（Spring+SpringMVC+MyBatis）**

2014-07-19作者：在路上来源：[在路上](http://www.350351.com/s.php?k=在路上) 4049次阅读

        使用SSM（Spring、SpringMVC和Mybatis）已经有三个多月了，项目在技术上已经没有什么难点了，基于现有的技术就可以实现想要的功能，当然肯定有很多可以改进的地方。之前没有记录SSM整合的过程，这次刚刚好基于自己的一个小项目重新搭建了一次，而且比项目搭建的要更好一些。以前解决问题的过程和方法并没有及时记录，以后在自己的小项目中遇到我再整理分享一下。这次，先说说三大框架整合过程。个人认为使用框架并不是很难，关键要理解其思想，这对于我们提高[编程](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%B1%E0%B3%CC&k0=%B1%E0%B3%CC&kdi0=8&k1=%B2%A9%BF%CD&kdi1=8&sid=d1228ebefca5db7&ch=0&tu=u1807396&jk=579145d909741a66&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)水平很有帮助。不过，如果用都不会，谈思想就变成纸上谈兵了！！！先技术，再思想。实践出真知。（可通过图片水印查看[博客](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%B2%A9%BF%CD&k0=%B1%E0%B3%CC&kdi0=8&k1=%B2%A9%BF%CD&kdi1=8&sid=d1228ebefca5db7&ch=0&tu=u1807396&jk=579145d909741a66&cf=29&fv=15&stid=9&urlid=0&luki=2&seller_id=1&di=128)地址）

1、基本概念

1.1、Spring

        Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson 在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。 简单来说，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。

1.2、SpringMVC

        Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。Spring MVC 分离了[控制器](http://baike.baidu.com/view/122229.htm)、模型[对象](http://baike.baidu.com/view/2387.htm)、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。

1.3、MyBatis

       MyBatis 本是[apache](http://baike.baidu.com/view/28283.htm)的一个开源项目[iBatis](http://baike.baidu.com/view/628102.htm), 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis 。MyBatis是一个基于Java的[持久层](http://baike.baidu.com/view/198047.htm)框架。iBATIS提供的[持久层](http://baike.baidu.com/view/198047.htm)框架包括SQL Maps和Data Access Objects（DAO）MyBatis 消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。

2、开发环境搭建

[**MyEclipse+Tomcat+MAVEN+SVN项目完整环境搭建**](http://blog.csdn.net/zhshulin/article/details/30779873)

分类： [开发环境搭建](http://blog.csdn.net/u012909091/article/category/2363697) 2014-06-14 20:57 1611人阅读 [评论](http://blog.csdn.net/zhshulin/article/details/30779873#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zhshulin/article/details/30779873#report)

[maven](http://www.csdn.net/tag/maven)[svn](http://www.csdn.net/tag/svn)[tomcat](http://www.csdn.net/tag/tomcat)[spket](http://www.csdn.net/tag/spket)[环境搭建](http://www.csdn.net/tag/%e7%8e%af%e5%a2%83%e6%90%ad%e5%bb%ba)

目录[(?)[+]](http://blog.csdn.net/zhshulin/article/details/30779873)

1. [JDK的安装](http://blog.csdn.net/zhshulin/article/details/30779873#t0)
2. [MyEclipse的安装](http://blog.csdn.net/zhshulin/article/details/30779873#t1)
3. [Tomcat8的安装](http://blog.csdn.net/zhshulin/article/details/30779873#t2)
4. [MyEclipse配置Tomcat](http://blog.csdn.net/zhshulin/article/details/30779873#t3)
5. [Maven安装配置](http://blog.csdn.net/zhshulin/article/details/30779873#t4)
   1. [maven配置环境变量](http://blog.csdn.net/zhshulin/article/details/30779873#t5)
   2. [Maven数据仓库的配置](http://blog.csdn.net/zhshulin/article/details/30779873#t6)
   3. [MyEclipse中集成Maven](http://blog.csdn.net/zhshulin/article/details/30779873#t7)
6. [SVN安装](http://blog.csdn.net/zhshulin/article/details/30779873#t8)
7. [JS代码提示安装Spket插件](http://blog.csdn.net/zhshulin/article/details/30779873#t9)
8. [JAVA代码提示](http://blog.csdn.net/zhshulin/article/details/30779873#t10)

       这次换了台电脑，所以需要重新配置一次项目开发环境，过程中的种种，记录下来，便于以后再次安装，同时给大家一个参考。

**1.JDK的安装**

       首先下载JDK，这个从sun公司官网可以下载，根据自己的系统选择64位还是32位，安装过程就是next一路到底。安装完成之后当然要配置环境变量了。

———————————————————————————————————————————————————

1.1新建变量名：JAVA\_HOME   变量值：E:\Java\jdk1.6.0\_43（这是我的jdk安装路径）

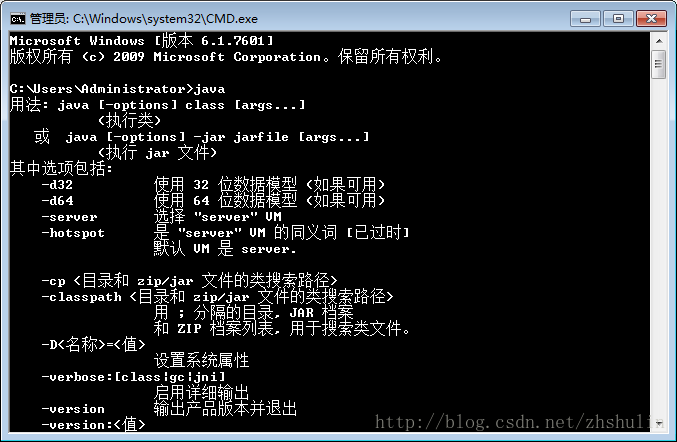
1.2编辑变量名：Path  在后面加上：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin

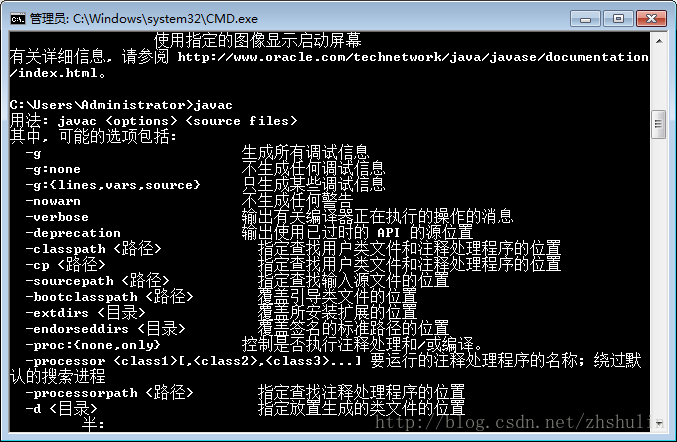
1.3 新建变量名：CLASSPATH 变量值： .;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar

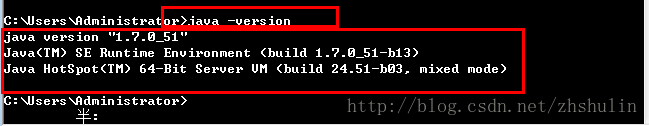
（注意：在设置变量的末尾时不要加上“；”）

———————————————————————————————————————————————————

设置完成之后我们当然需要测试，打开“运行”-->输入“CMD"-->在命令行上面输入：java，然后回车；输入javac，回车，输入java -version，回车，如果出现下列三个画面，那么JDK配置成功，否则就需要检查环境变量的配置。

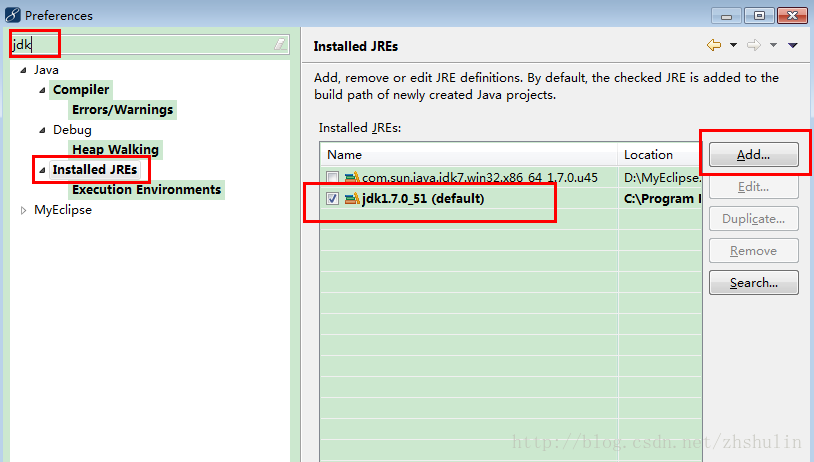






**2.MyEclipse的安装**

       我是用的是MyEclipse2014，这个大家下载了之后也是一路安装到底，中间可以选择是安装32位还是64位。完成之后我们启动MyEclipse，设置好工作路径，也可以直接默认。我们打开window-->preference输入jdk，安装下图，选择Eclipse默认的JRE为自己安装的，而不是eclipse自带的。



       此时我们可以随便建立一个JAVA项目，然后编写一个helloworld来测试一下是否正确。

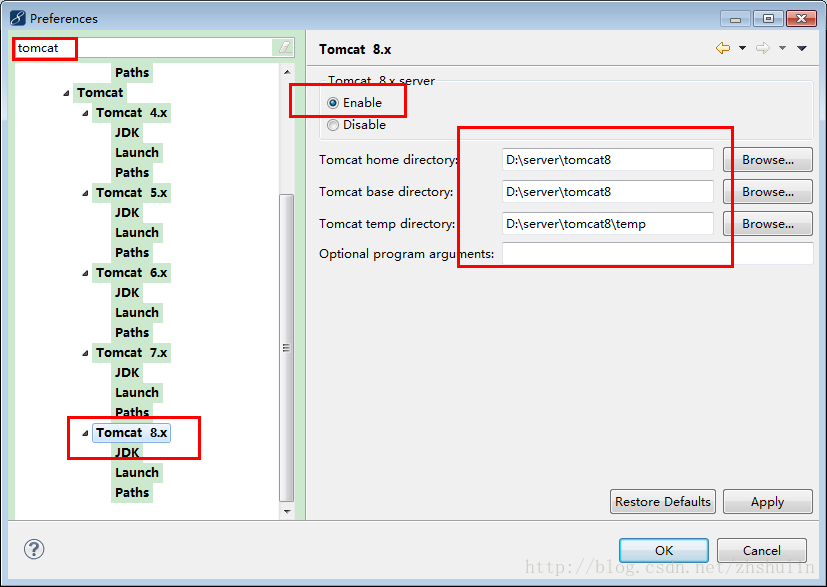
**3.Tomcat8的安装**

       Tomcat我使用的版本是8，大家可以到apache的官网上去下载，<http://tomcat.apache.org/download-80.cgi>，我是用的安装版，所以不需要配置任何东西，如果人比较懒不妨就下个安装版吧。压缩包版本的需要自己配置环境变量，这个问下度娘就可以了，我本人比较懒...见谅！

       安装版本的安装过程也是一路到底，而且不需要配置环境变量。

**4.MyEclipse配置Tomcat**

       项目开发的时候最好不要是用MyEclipse自带的Tomcat，所以需要配置我们安装好的tomcat。打开window-->preference输入tomcat，按照下图进行配置即可，其他版本的tomcat对应着配置就可以了。



**5.Maven安装配置**

**5.1 maven配置环境变量**

 下载maven，就是一个包：apache-maven-3.2.1，我用的版本是3.2.1，解压到一个路径，然后配置环境变量：

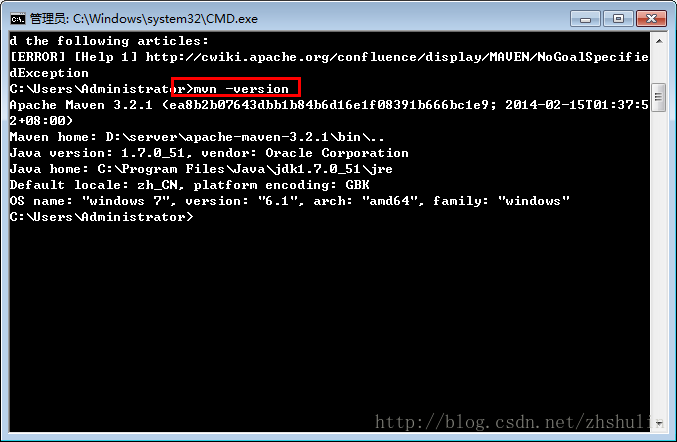
——————————————————————————————————————————————————

      5.1 新建变量名：MAVEN  变量值：D:\server\apache-maven-3.2.1（这是我的MAVEN路径）

      5.2 编辑变量名：Path  在最前面加上：%MAVEN\_HOME%\bin;（注意，最后要有个";"作为分隔符）

——————————————————————————————————————————————————

完成之后，在命令行输入：mvn -version查看是否有以下内容，如果有表示配置成功。

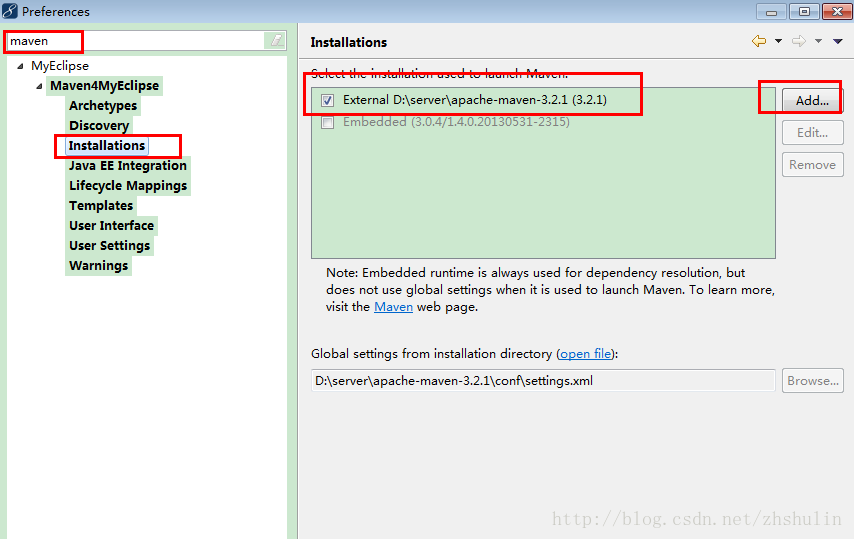


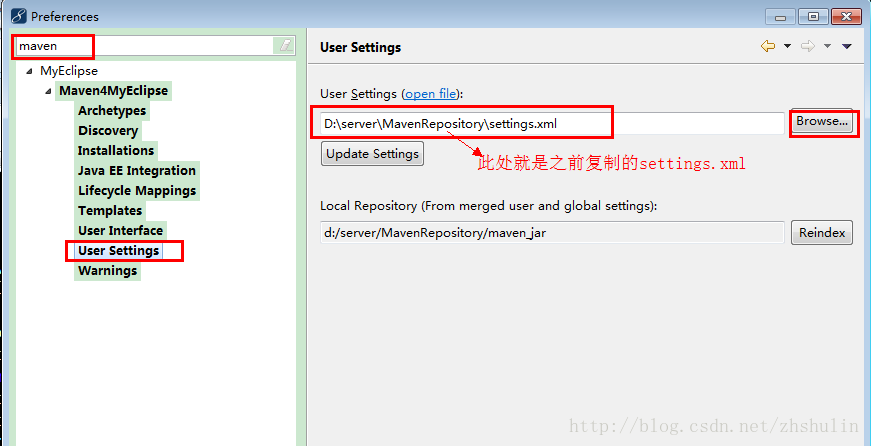
**5.2Maven数据仓库的配置**

        MAVEN中还有一个重要的配置就是数据仓库的路径配置，我们找到MAVEN的安装路径，进入conf-->打开settings.xml，找到localRepository标签，此时是被注释掉的，我们解除注释，然后配置一个路径，例如：d:/server/MavenRepository/maven\_jar，这样以后MAVEN管理下载的jar包都会在这个路径下。当然我们需要建这样一个目录结构，然后还要讲settings.xml复制一份到d:/server/MavenRepository下，这个在与MyEclipse集成时会用到。

**5.3MyEclipse中集成Maven**

      一样在preferences中进行配置，如下图：





       此时，点击File-->new-->other-->MyEclipse-->Maven4MyEclipse-->Maven Project，如果可以看到这些，证明配置成功，已经可以建立Maven项目了。

**6.SVN安装**

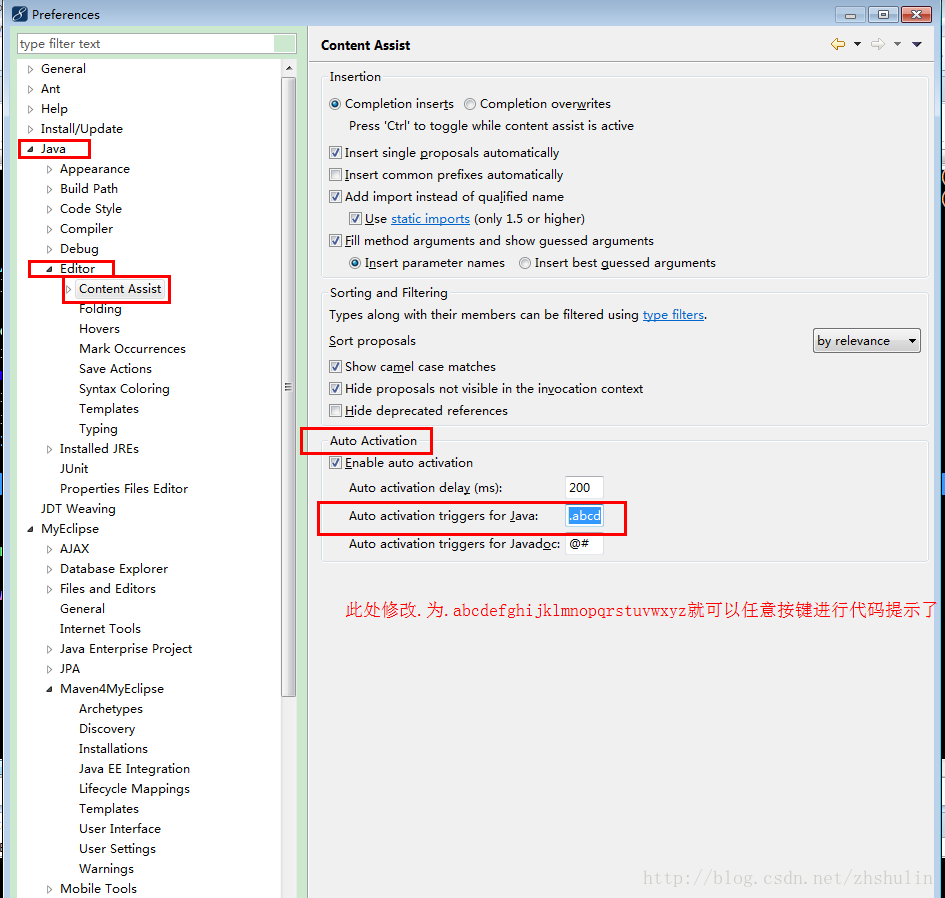
     SVN是代码版本管理器，首先在本地安装一个SVN管理器，然后配置MyEclipse的SVN插件，将插件下载下来解压后，直接复制到MyEclipse安装目录下dropins中即可。任意建立一个项目，右键项目-->team-->share Project..，打开后如果看到了SVN代表插件安装成功。

**7. JS代码提示安装（Spket插件）**

     参看之前的一篇博文：<http://blog.csdn.net/zhshulin/article/details/25635883>

**8. JAVA代码提示**

       经过以上过程，整个项目需要的环境差不多搭建完成了。接下来一个小技巧，如下图进行配置之后就可以将只在.出现时进行代码提示换成任意字母+.出现时的代码提示了（.abcdefghijklmnopqrstuvwxyz）：



3、Maven Web项目创建

[**使用maven创建web项目**](http://blog.csdn.net/zhshulin/article/details/37921705)

分类： [J2EE](http://blog.csdn.net/u012909091/article/category/2487773) 2014-07-18 09:36 1059人阅读 [评论](http://blog.csdn.net/zhshulin/article/details/37921705#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zhshulin/article/details/37921705#report)

[maven](http://www.csdn.net/tag/maven)[java](http://www.csdn.net/tag/java)[web](http://www.csdn.net/tag/web)[项目创建](http://www.csdn.net/tag/%e9%a1%b9%e7%9b%ae%e5%88%9b%e5%bb%ba)

        目前做的项目使用的是MAVEN来管理jar包，这也是我第一次接触maven，感觉非常好，再也不用一个一个去添加和下载jar包了，直接在maven配置文件中配置就可以了，maven可以帮助我们自动下载。非常方便。之前比较忙没时间整理，现在整理一下，记录下来，加深印象。当然我们使用maven也只是最基本的，很多其他功能都没有使用到，本篇仅介绍如何创建一个maven项目，不做其他。

       环境搭建参考之前一篇博文：http://blog.csdn.net/zhshulin/article/details/30779873

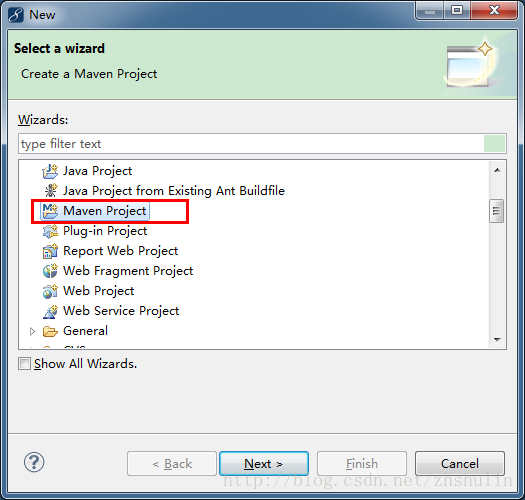
        下面推荐2个maven找jar包配置的网站，只需要搜索关键字即可找到需要的Jar包，非常方便，比如：mysql就可以找到mysql-connect-java.jar。

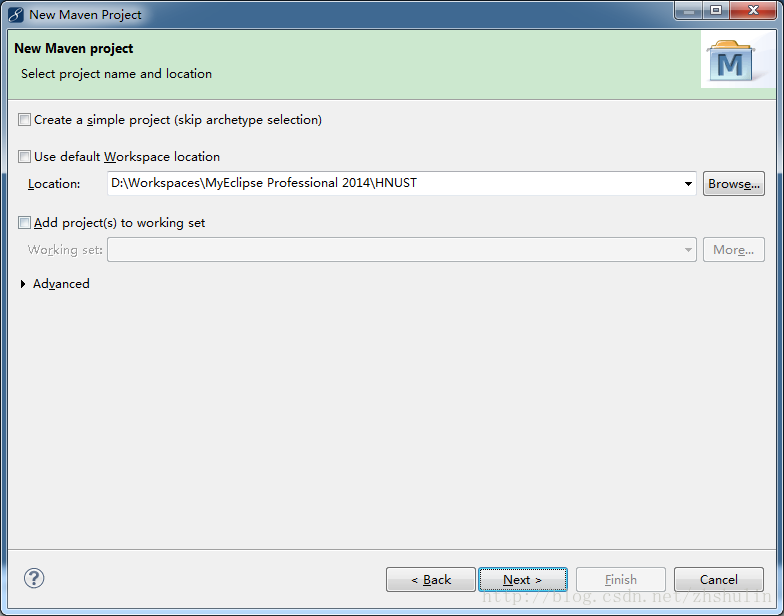
         http://search.maven.org/#search%7Cga%7C1%7Cmybatis

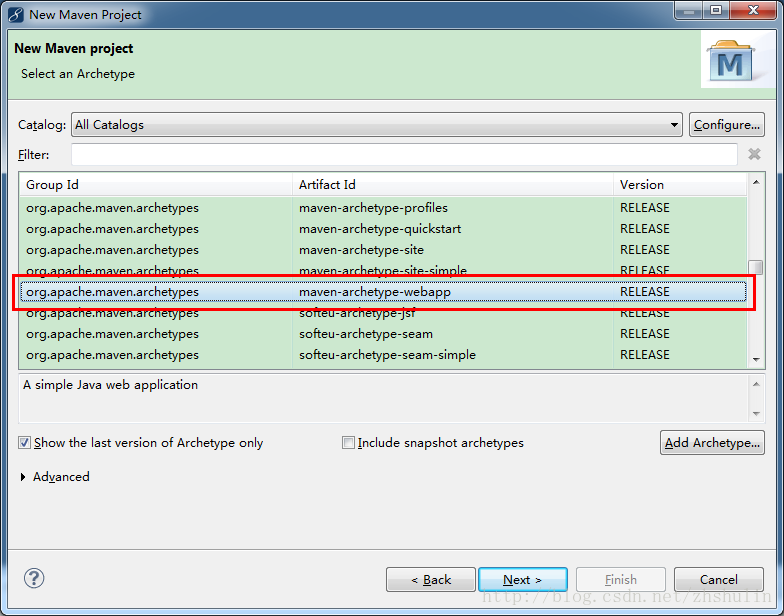
         http://mvnrepository.com/

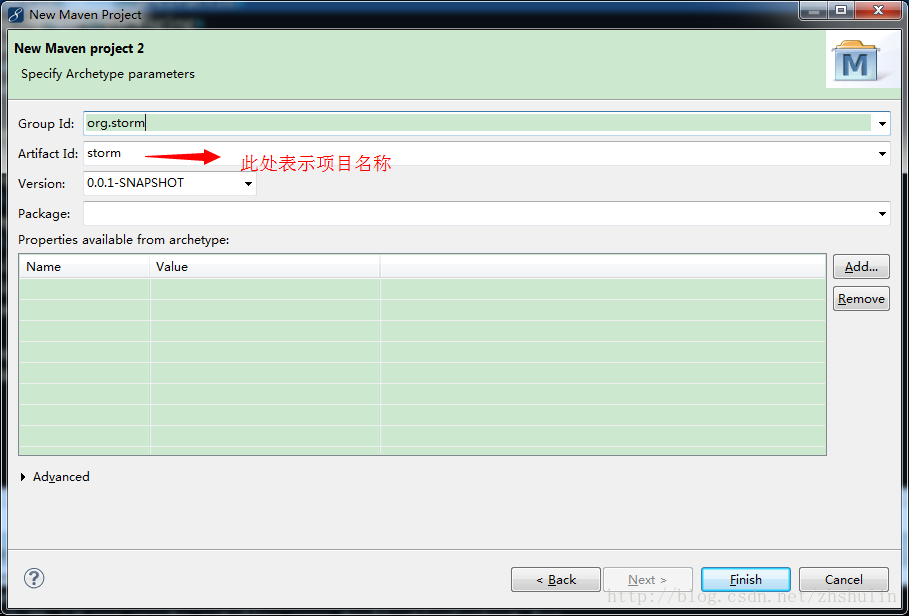
下面是创建的相关图文教程：

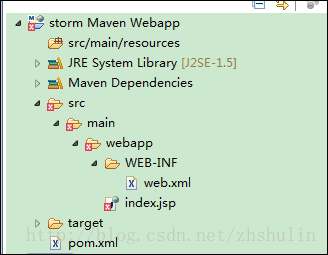
1、首先新建一个maven项目，看图：





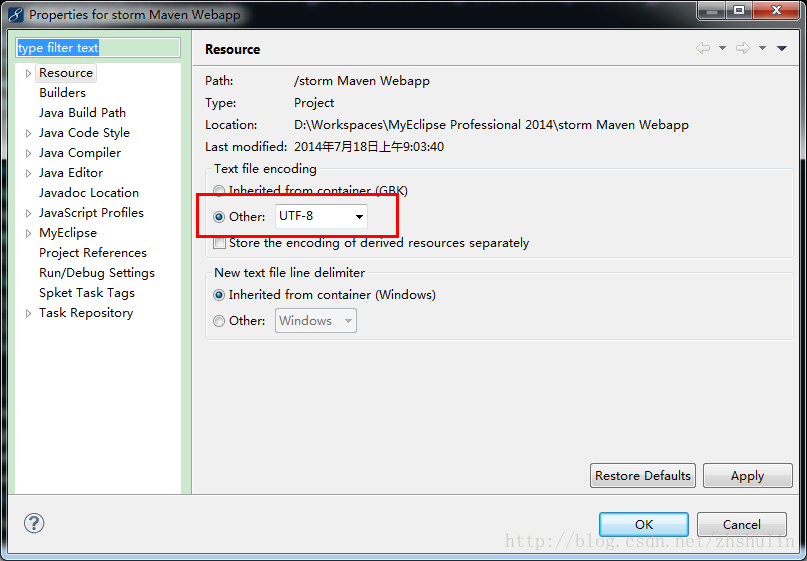




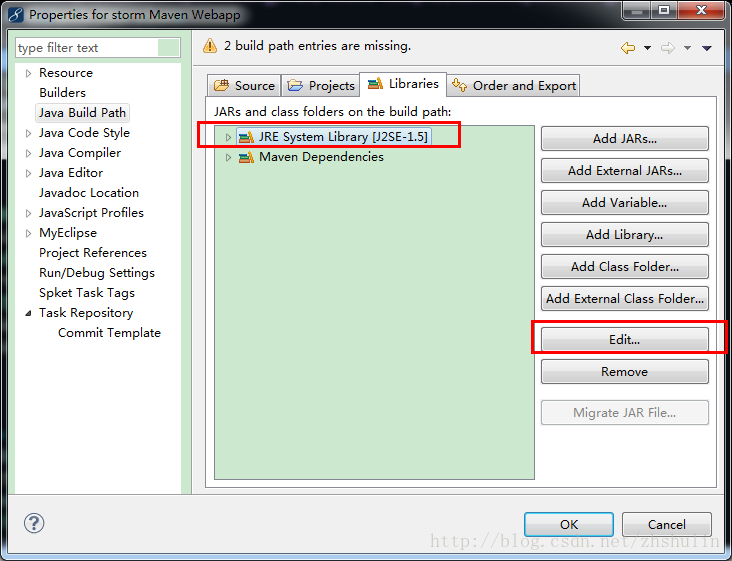


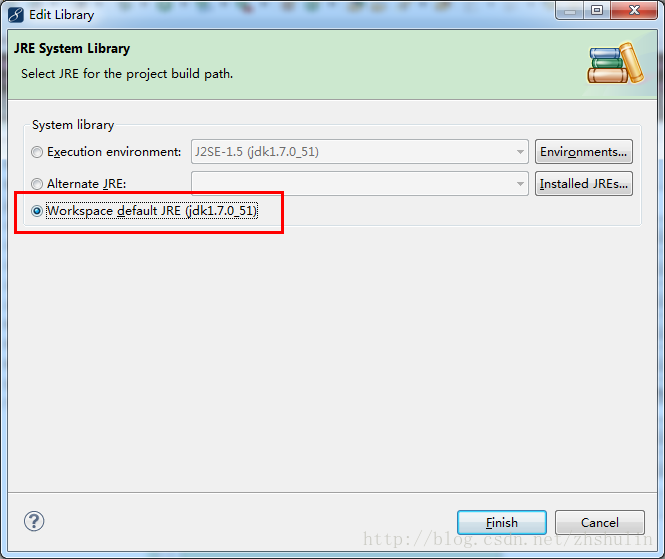
2、按照以上步骤就可以创建一个maven项目，可以看到最下图的目录结构，但是这样的目录结构是不对的，需要做一些修改。

首先为了避免乱码，我们应该将项目编码换成UTF-8，以后项目编码都采用UTF-8，加个字符编码过滤器，乱码就不再“乱”了（不表示没有，但是比较容易解决）。



        再次，按照以下图解来进行操作（设置JRE为默认的），这样就会出现完整的、正确的目录结构，至于什么原因，我不是很清楚，应该是个BUG：





        最后，还有个index.jsp报错，我们需要引入java ee的jar包即可。在pom.xml中引入：

**[html]** [view plaincopyprint?](http://blog.csdn.net/zhshulin/article/details/37921705)

1. <!-- 导入java ee jar 包 -->
2. **<dependency>**
3. **<groupId>**javax**</groupId>**
4. **<artifactId>**javaee-api**</artifactId>**
5. **<version>**7.0**</version>**
6. **</dependency>**

<!-- 导入java ee jar 包 -->

<dependency>

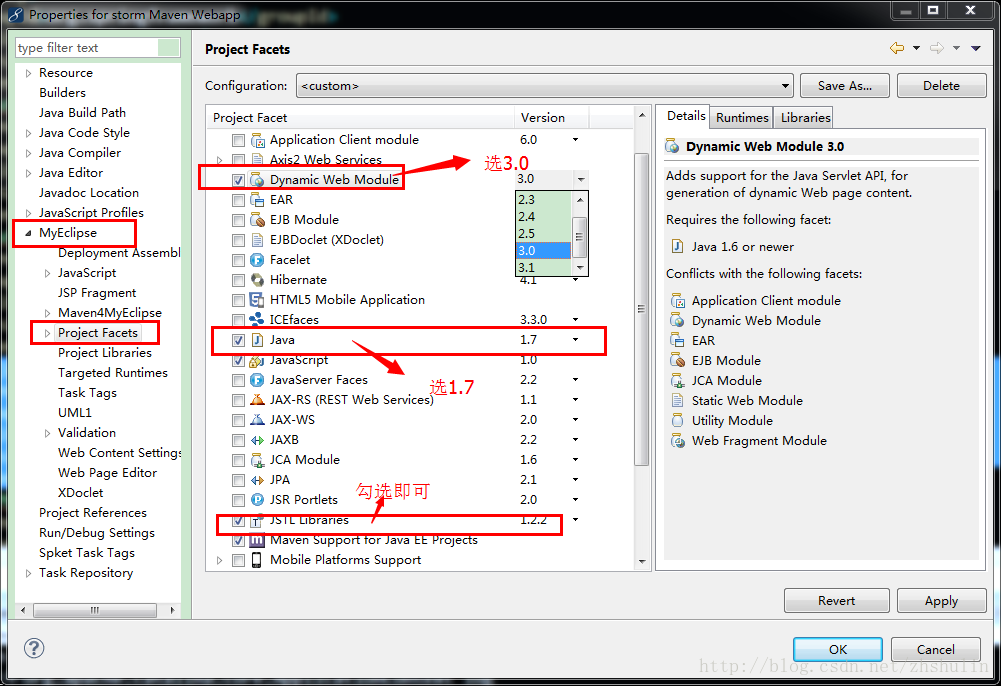
<groupId>javax</groupId>

<artifactId>javaee-api</artifactId>

<version>7.0</version>

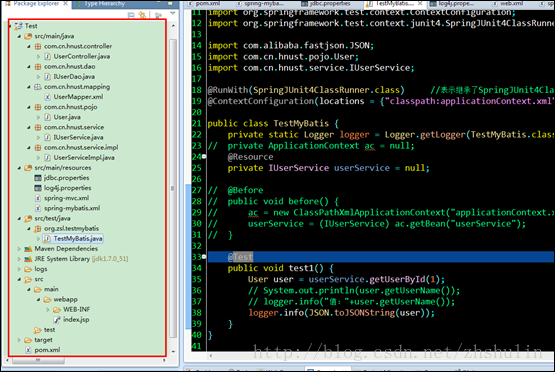
</dependency>

        至此，完整的目录结构以及正确的项目搞定了。接下来再进行一些配置，毕竟是web项目，需要jstl等等，配置如下：



4、SSM整合

        下面主要介绍三大框架的整合，至于环境的搭建以及项目的创建，参看上面的博文。这次整合我分了2个配置文件，分别是spring-mybatis.xml，包含spring和mybatis的配置文件，还有个是spring-mvc的配置文件，此外有2个资源文件：jdbc.propertis和log4j.properties。完整目录结构如下：



使用框架都是较新的版本：

       Spring 4.0.2 RELEASE

       Spring MVC 4.0.2 RELEASE

       MyBatis 3.2.6

4.1、Maven引入需要的JAR包

         为了方便后面说的时候不需要引入JAR包，我这里直接给出所有需要的JAR包，这都是基本的JAR包，每个包的是干什么的都有注释，就不再多说了。

pom.xml

<properties>

<!-- [spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)版本号 -->

<spring.version>4.0.2.RELEASE</spring.version>

<!-- mybatis版本号 -->

<mybatis.version>3.2.6</mybatis.version>

<!-- log4j日志文件管理包版本 -->

<slf4j.version>1.7.7</slf4j.version>

<log4j.version>1.2.17</log4j.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<!-- 表示开发的时候引入，发布的时候不会加载此包 -->

<scope>test</scope>

</dependency>

<!-- [spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)核心包 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-oxm</artifactId>

<version>${[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128).version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- mybatis核心包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>${mybatis.version}</version>

</dependency>

<!-- mybatis/spring包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<!-- 导入[java](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=java&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=2&seller_id=1&di=128) ee jar 包 -->

<dependency>

<groupId>javax</groupId>

<artifactId>javaee-api</artifactId>

<version>7.0</version>

</dependency>

<!-- 导入Mysql数据库链接jar包 -->

<dependency>

<groupId>[mysql](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=mysql&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=4&seller_id=1&di=128)</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

<!-- 导入dbcp的jar包，用来在applicationContext.xml中配置[数据库](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%CA%FD%BE%DD%BF%E2&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128) -->

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>

<!-- JSTL标签类 -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- 日志文件管理包 -->

<!-- log start -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>${log4j.version}</version>

</dependency>

<!-- 格式化对象，方便输出日志 -->

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.1.41</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${slf4j.version}</version>

</dependency>

<!-- log end -->

<!-- 映入JSON -->

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>

</dependency>

<!-- 上传[组件](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F3%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%D7%E9%BC%FE&k0=spring&kdi0=8&k1=java&kdi1=8&k2=%CA%FD%BE%DD%BF%E2&kdi2=8&k3=mysql&kdi3=8&k4=%D7%E9%BC%FE&kdi4=8&k5=%B0%E6%B1%BE%BA%C5&kdi5=8&sid=2fc2ab6c691db298&ch=0&tu=u1807396&jk=aaecf05aa7b9297b&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)包 -->

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.3.1</version>

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.4</version>

</dependency>

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

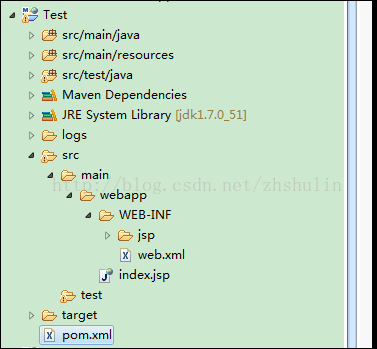
<version>1.9</version>

</dependency>

</dependencies>

4.2、Spring与MyBatis的整合

所有需要的JAR包都引入以后，首先进行Spring与MyBatis的整合，然后再进行JUnit测试，先看一个项目结构图：



4.2.1、建立JDBC属性文件

jdbc.properties（文件编码修改为utf-8）

driver=com.mysql.jdbc.Driver

url=jdbc:mysql://10.221.10.111:8080/db\_zsl

username=demao

password=demao

#定义初始连接数

initialSize=0

#定义最大连接数

maxActive=20

#定义最大空闲

maxIdle=20

#定义最小空闲

minIdle=1

#定义最长等待时间

maxWait=60000

4.2.2、建立spring-mybatis.xml配置文件

        这个文件就是用来完成spring和mybatis的整合的。这里面也没多少行配置，主要的就是自动扫描，自动注入，配置数据库。注释也很详细，大家看看就明白了。

spring-mybatis.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd">

<!-- 自动扫描 -->

<context:component-scan base-package="com.cn.hnust" />

<!-- 引入配置文件 -->

<bean id="propertyConfigurer"

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="location" value="classpath:jdbc.properties" />

</bean>

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"

destroy-method="close">

<property name="driverClassName" value="${driver}" />

<property name="url" value="${url}" />

<property name="username" value="${username}" />

<property name="password" value="${password}" />

<!-- 初始化连接大小 -->

<property name="initialSize" value="${initialSize}"></property>

<!-- 连接池最大数量 -->

<property name="maxActive" value="${maxActive}"></property>

<!-- 连接池最大空闲 -->

<property name="maxIdle" value="${maxIdle}"></property>

<!-- 连接池最小空闲 -->

<property name="minIdle" value="${minIdle}"></property>

<!-- 获取连接最大等待时间 -->

<property name="maxWait" value="${maxWait}"></property>

</bean>

<!-- spring和MyBatis完美整合，不需要mybatis的配置映射文件 -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource" />

<!-- 自动扫描mapping.xml文件 -->

<property name="mapperLocations" value="classpath:com/cn/hnust/mapping/\*.xml"></property>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="com.cn.hnust.dao" />

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"></property>

</bean>

<!-- (事务管理)transaction manager, use JtaTransactionManager for global tx -->

<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<property name="dataSource" ref="dataSource" />

</bean>

</beans>

4.2.3、Log4j的配置

      为了方便调试，一般都会使用日志来输出信息，Log4j是[Apache](http://baike.baidu.com/subview/28283/5418753.htm)的一个开放[源代码](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F5%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%D4%B4%B4%FA%C2%EB&k0=%D4%B4%B4%FA%C2%EB&kdi0=1&k1=gui&kdi1=8&k2=%D7%E9%BC%FE&kdi2=8&k3=unix&kdi3=8&k4=%B2%A9%CE%C4&kdi4=8&sid=555c0b11d0b7424&ch=0&tu=u1807396&jk=b918d45c720bb8e2&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)项目，通过使用Log4j，我们可以控制[日志](http://baike.baidu.com/view/269351.htm)信息输送的目的地是[控制台](http://baike.baidu.com/view/135167.htm)、文件、[GUI](http://baike.baidu.com/view/25309.htm)[组件](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F5%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%D7%E9%BC%FE&k0=%D4%B4%B4%FA%C2%EB&kdi0=1&k1=gui&kdi1=8&k2=%D7%E9%BC%FE&kdi2=8&k3=unix&kdi3=8&k4=%B2%A9%CE%C4&kdi4=8&sid=555c0b11d0b7424&ch=0&tu=u1807396&jk=b918d45c720bb8e2&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)，甚至是套接口服务器、[NT](http://baike.baidu.com/subview/378/5113303.htm)的事件记录器、[UNIX](http://baike.baidu.com/view/8095.htm) [Syslog](http://baike.baidu.com/view/1614723.htm)[守护进程](http://baike.baidu.com/view/53123.htm)等；我们也可以控制每一条日志的输出格式；通过定义每一条日志信息的级别，我们能够更加细致地控制日志的生成过程。

      Log4j的配置很简单，而且也是通用的，下面给出一个基本的配置，换到其他项目中也无需做多大的调整，如果想做调整或者想了解Log4j的各种配置，参看我转载的一篇[博文](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F5%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%B2%A9%CE%C4&k0=%D4%B4%B4%FA%C2%EB&kdi0=1&k1=gui&kdi1=8&k2=%D7%E9%BC%FE&kdi2=8&k3=unix&kdi3=8&k4=%B2%A9%CE%C4&kdi4=8&sid=555c0b11d0b7424&ch=0&tu=u1807396&jk=b918d45c720bb8e2&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)，很详细：

[**Log4j配置详解**](http://blog.csdn.net/zhshulin/article/details/37937365)

分类： [JAVA](http://blog.csdn.net/u012909091/article/category/1835109) 2014-07-18 21:07 304人阅读 [评论](http://blog.csdn.net/zhshulin/article/details/37937365#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zhshulin/article/details/37937365#report)

[log4j](http://www.csdn.net/tag/log4j)[配置](http://www.csdn.net/tag/%e9%85%8d%e7%bd%ae)

来自: <http://www.blogjava.net/zJun/archive/2006/06/28/55511.html>

Log4J的配置文件(Configuration File)就是用来设置记录器的级别、存放器和布局的，它可接key=value格式的设置或xml格式的设置信息。通过配置，可以创建出Log4J的运行环境。  
  
**1. 配置文件**  
Log4J配置文件的基本格式如下：

http://www.blogjava.net/Images/OutliningIndicators/None.gif#配置根Logger  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.rootLogger  =   **[** level **]**   ,  appenderName1 ,  appenderName2 ,  …  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif#配置日志信息输出目的地Appender  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.appenderName  =  fully.qualified.name.of.appender.class   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　log4j.appender.appenderName.option1  =  value1   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　…   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　log4j.appender.appenderName.optionN  =  valueN   
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif#配置日志信息的格式（布局）  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.appenderName.layout  =  fully.qualified.name.of.layout.class   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　log4j.appender.appenderName.layout.option1  =  value1   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　…   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　log4j.appender.appenderName.layout.optionN  =  valueN

其中 **[level]**是日志输出级别，共有5级：

**http://www.blogjava.net/Images/OutliningIndicators/None.gifFATAL       0    
http://www.blogjava.net/Images/OutliningIndicators/None.gifERROR      3    
http://www.blogjava.net/Images/OutliningIndicators/None.gifWARN       4    
http://www.blogjava.net/Images/OutliningIndicators/None.gifINFO          6    
http://www.blogjava.net/Images/OutliningIndicators/None.gifDEBUG      7   
http://www.blogjava.net/Images/OutliningIndicators/None.gif**

Appender 为日志输出目的地，Log4j提供的appender有以下几种：

http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.ConsoleAppender（控制台），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.FileAppender（文件），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.DailyRollingFileAppender（每天产生一个日志文件），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.RollingFileAppender（文件大小到达指定尺寸的时候产生一个新的文件），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.WriterAppender（将日志信息以流格式发送到任意指定的地方）

**Layout**：日志输出格式，Log4j提供的layout有以下几种：

http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.HTMLLayout（以HTML表格形式布局），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.PatternLayout（可以灵活地指定布局模式），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.SimpleLayout（包含日志信息的级别和信息字符串），  
http://www.blogjava.net/Images/OutliningIndicators/None.giforg.apache.log4j.TTCCLayout（包含日志产生的时间、线程、类别等等信息）

**打印参数:**Log4J采用类似C语言中的printf函数的打印格式格式化日志信息，如下:

http://www.blogjava.net/Images/OutliningIndicators/None.gif 　  **%m**   输出代码中指定的消息  
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%p**   输出优先级，即DEBUG，INFO，WARN，ERROR，FATAL   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%r**   输出自应用启动到输出该log信息耗费的毫秒数   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%c**   输出所属的类目，通常就是所在类的全名   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%t**   输出产生该日志事件的线程名   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%n**  输出一个回车换行符，Windows平台为“/r/n”，Unix平台为“/n”   
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%d**   输出日志时间点的日期或时间，默认格式为ISO8601，也可以在其后指定格式，比如：%d{yyy MMM dd HH:mm:ss ,SSS}，输出类似：2002年10月18日  22 ： 10 ： 28 ， 921    
http://www.blogjava.net/Images/OutliningIndicators/None.gif　　**%l**  输出日志事件的发生位置，包括类目名、发生的线程，以及在代码中的行数。举例：Testlog4.main(TestLog4.java: 10 ) http://www.blogjava.net/Images/OutliningIndicators/None.gif

**2. 在代码中初始化Logger:**   
1）在程序中调用**BasicConfigurator.configure()**方法：给根记录器增加一个ConsoleAppender，输出格式通过PatternLayout设为**"%-4r [%t] %-5p %c %x - %m%n"**，还有根记录器的默认级别是**Level.DEBUG**.   
2）配置放在文件里，通过命令行参数传递文件名字，通过**PropertyConfigurator.configure(args[x])**解析并配置；  
3）配置放在文件里，通过环境变量传递文件名等信息，利用log4j默认的初始化过程解析并配置；  
4）配置放在文件里，通过应用服务器配置传递文件名等信息，利用一个特殊的servlet来完成配置。  
  
**3. 为不同的 Appender 设置日志输出级别：**  
当调试系统时，我们往往注意的只是异常级别的日志输出，但是通常所有级别的输出都是放在一个文件里的，如果日志输出的级别是BUG！？那就慢慢去找吧。  
这时我们也许会想要是能把异常信息单独输出到一个文件里该多好啊。当然可以，Log4j已经提供了这样的功能，我们只需要在配置中修改**Appender**的**Threshold** 就能实现,比如下面的例子：  
  
**[配置文件]**

http://www.blogjava.net/Images/OutliningIndicators/None.gif ### set log levels ###  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.rootLogger = debug ,  stdout ,  D ,  E  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif### 输出到控制台 ###  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.stdout = org.apache.log4j.ConsoleAppender  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.stdout.Target = System.out  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.stdout.layout = org.apache.log4j.PatternLayout  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.stdout.layout.ConversionPattern =  %d{ABSOLUTE} %5p %c{ 1 }:%L - %m%n  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif### 输出到日志文件 ###  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D = org.apache.log4j.DailyRollingFileAppender  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.File = logs/log.log  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.Append = true  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.**Threshold** = DEBUG **## 输出DEBUG级别以上的日志**  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.layout = org.apache.log4j.PatternLayout  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss}  **[** %t:%r **]** - **[** %p **]**  %m%n  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif### 保存异常信息到单独文件 ###  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D = org.apache.log4j.DailyRollingFileAppender  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.File = logs/error.log ## 异常日志文件名  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.Append = true  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.**Threshold** = ERROR**## 只输出ERROR级别以上的日志!!!**http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.layout = org.apache.log4j.PatternLayout  
http://www.blogjava.net/Images/OutliningIndicators/None.giflog4j.appender.D.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss}  **[** %t:%r **]** - **[** %p **]**  %m%n

**[代码中使用]**

http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gif public   class  TestLog4j  {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif     public   static   void  main(String[] args)  {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        PropertyConfigurator.configure( " D:/Code/conf/log4j.properties " );  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        Logger logger  =  Logger.getLogger(TestLog4j. class );  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        logger.debug( " debug " );  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        logger.error( " error " );  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }   
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}

运行一下，看看异常信息是不是保存在了一个单独的文件error.log中

log4j.properties 使用  
一.参数意义说明  
输出级别的种类  
ERROR、WARN、INFO、DEBUG  
ERROR 为严重错误 主要是程序的错误  
WARN 为一般警告，比如session丢失  
INFO 为一般要显示的信息，比如登录登出  
DEBUG 为程序的调试信息  
配置日志信息输出目的地  
log4j.appender.appenderName = fully.qualified.name.of.appender.class  
1.org.apache.log4j.ConsoleAppender（控制台）  
2.org.apache.log4j.FileAppender（文件）  
3.org.apache.log4j.DailyRollingFileAppender（每天产生一个日志文件）  
4.org.apache.log4j.RollingFileAppender（文件大小到达指定尺寸的时候产生一个新的文件）  
5.org.apache.log4j.WriterAppender（将日志信息以流格式发送到任意指定的地方）  
配置日志信息的格式  
log4j.appender.appenderName.layout = fully.qualified.name.of.layout.class  
1.org.apache.log4j.HTMLLayout（以HTML表格形式布局），  
2.org.apache.log4j.PatternLayout（可以灵活地指定布局模式），  
3.org.apache.log4j.SimpleLayout（包含日志信息的级别和信息字符串），  
4.org.apache.log4j.TTCCLayout（包含日志产生的时间、线程、类别等等信息）  
控制台选项  
Threshold=DEBUG:指定日志消息的输出最低层次。  
ImmediateFlush=true:默认值是true,意谓着所有的消息都会被立即输出。  
Target=System.err：默认情况下是：System.out,指定输出控制台  
FileAppender 选项  
Threshold=DEBUF:指定日志消息的输出最低层次。  
ImmediateFlush=true:默认值是true,意谓着所有的消息都会被立即输出。  
File=mylog.txt:指定消息输出到mylog.txt文件。  
Append=false:默认值是true,即将消息增加到指定文件中，false指将消息覆盖指定的文件内容。  
RollingFileAppender 选项  
Threshold=DEBUG:指定日志消息的输出最低层次。  
ImmediateFlush=true:默认值是true,意谓着所有的消息都会被立即输出。  
File=mylog.txt:指定消息输出到mylog.txt文件。  
Append=false:默认值是true,即将消息增加到指定文件中，false指将消息覆盖指定的文件内容。  
MaxFileSize=100KB: 后缀可以是KB, MB 或者是 GB. 在日志文件到达该大小时，将会自动滚动，即将原来的内容移到mylog.log.1文件。  
MaxBackupIndex=2:指定可以产生的滚动文件的最大数。  
log4j.appender.A1.layout.ConversionPattern=%-4r %-5p %d{yyyy-MM-dd HH:mm:ssS} %c %m%n  
日志信息格式中几个符号所代表的含义：  
 -X号: X信息输出时左对齐；  
 %p: 输出日志信息优先级，即DEBUG，INFO，WARN，ERROR，FATAL,  
 %d: 输出日志时间点的日期或时间，默认格式为ISO8601，也可以在其后指定格式，比如：%d{yyy MMM dd HH:mm:ss,SSS}，输出类似：2002年10月18日 22：10：28，921  
 %r: 输出自应用启动到输出该log信息耗费的毫秒数  
 %c: 输出日志信息所属的类目，通常就是所在类的全名  
 %t: 输出产生该日志事件的线程名  
 %l: 输出日志事件的发生位置，相当于%C.%M(%F:%L)的组合,包括类目名、发生的线程，以及在代码中的行数。举例：Testlog4.main (TestLog4.java:10)  
 %x: 输出和当前线程相关联的NDC(嵌套诊断环境),尤其用到像java servlets这样的多客户多线程的应用中。  
 %%: 输出一个"%"字符  
 %F: 输出日志消息产生时所在的文件名称  
 %L: 输出代码中的行号  
 %m: 输出代码中指定的消息,产生的日志具体信息  
 %n: 输出一个回车换行符，Windows平台为"/r/n"，Unix平台为"/n"输出日志信息换行  
 可以在%与模式字符之间加上修饰符来控制其最小宽度、最大宽度、和文本的对齐方式。如：  
 1)%20c：指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，默认的情况下右对齐。  
 2)%-20c:指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，"-"号指定左对齐。  
 3)%.30c:指定输出category的名称，最大的宽度是30，如果category的名称大于30的话，就会将左边多出的字符截掉，但小于30的话也不会有空格。  
 4)%20.30c:如果category的名称小于20就补空格，并且右对齐，如果其名称长于30字符，就从左边较远输出的字符截掉。  
二.文件配置Sample1  
log4j.rootLogger=DEBUG,A1,R  
#log4j.rootLogger=INFO,A1,R  
# ConsoleAppender 输出  
log4j.appender.A1=org.apache.log4j.ConsoleAppender  
log4j.appender.A1.layout=org.apache.log4j.PatternLayout  
log4j.appender.A1.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss,SSS} [%c]-[%p] %m%n  
# File 输出 一天一个文件,输出路径可以定制,一般在根路径下  
log4j.appender.R=org.apache.log4j.DailyRollingFileAppender  
log4j.appender.R.File=blog\_log.txt  
log4j.appender.R.MaxFileSize=500KB  
log4j.appender.R.MaxBackupIndex=10  
log4j.appender.R.layout=org.apache.log4j.PatternLayout  
log4j.appender.R.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss,SSS} [%t] [%c] [%p] - %m%n  
文件配置Sample2  
下面给出的Log4J配置文件实现了输出到控制台，文件，回滚文件，发送日志邮件，输出到数据库日志表，自定义标签等全套功能。  
log4j.rootLogger=DEBUG,CONSOLE,A1,im   
#DEBUG,CONSOLE,FILE,ROLLING\_FILE,MAIL,DATABASE  
log4j.addivity.org.apache=true  
###################   
# Console Appender   
###################   
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender   
log4j.appender.Threshold=DEBUG   
log4j.appender.CONSOLE.Target=System.out   
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout   
log4j.appender.CONSOLE.layout.ConversionPattern=[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n   
#log4j.appender.CONSOLE.layout.ConversionPattern=[start]%d{DATE}[DATE]%n%p[PRIORITY]%n%x[NDC]%n%t[THREAD] n%c[CATEGORY]%n%m[MESSAGE]%n%n  
#####################   
# File Appender   
#####################   
log4j.appender.FILE=org.apache.log4j.FileAppender   
log4j.appender.FILE.File=file.log   
log4j.appender.FILE.Append=false   
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout   
log4j.appender.FILE.layout.ConversionPattern=[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n   
# Use this layout for LogFactor 5 analysis  
########################   
# Rolling File   
########################   
log4j.appender.ROLLING\_FILE=org.apache.log4j.RollingFileAppender   
log4j.appender.ROLLING\_FILE.Threshold=ERROR   
log4j.appender.ROLLING\_FILE.File=rolling.log   
log4j.appender.ROLLING\_FILE.Append=true   
log4j.appender.ROLLING\_FILE.MaxFileSize=10KB   
log4j.appender.ROLLING\_FILE.MaxBackupIndex=1   
log4j.appender.ROLLING\_FILE.layout=org.apache.log4j.PatternLayout   
log4j.appender.ROLLING\_FILE.layout.ConversionPattern=[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n  
####################   
# Socket Appender   
####################   
log4j.appender.SOCKET=org.apache.log4j.RollingFileAppender   
log4j.appender.SOCKET.RemoteHost=localhost   
log4j.appender.SOCKET.Port=5001   
log4j.appender.SOCKET.LocationInfo=true   
# Set up for Log Facter 5   
log4j.appender.SOCKET.layout=org.apache.log4j.PatternLayout   
log4j.appender.SOCET.layout.ConversionPattern=[start]%d{DATE}[DATE]%n%p[PRIORITY]%n%x[NDC]%n%t[THREAD]%n%c[CATEGORY]%n%m[MESSAGE]%n%n  
########################   
# Log Factor 5 Appender   
########################   
log4j.appender.LF5\_APPENDER=org.apache.log4j.lf5.LF5Appender   
log4j.appender.LF5\_APPENDER.MaxNumberOfRecords=2000  
########################   
# SMTP Appender   
#######################   
log4j.appender.MAIL=org.apache.log4j.net.SMTPAppender   
log4j.appender.MAIL.Threshold=FATAL   
log4j.appender.MAIL.BufferSize=10   
[log4j.appender.MAIL.From=chenyl@yeqiangwei.com](mailto:log4j.appender.MAIL.From=chenyl@yeqiangwei.com)  
log4j.appender.MAIL.SMTPHost=mail.hollycrm.com   
log4j.appender.MAIL.Subject=Log4J Message   
[log4j.appender.MAIL.To=chenyl@yeqiangwei.com](mailto:log4j.appender.MAIL.To=chenyl@yeqiangwei.com)  
log4j.appender.MAIL.layout=org.apache.log4j.PatternLayout   
log4j.appender.MAIL.layout.ConversionPattern=[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n  
########################   
# JDBC Appender   
#######################   
log4j.appender.DATABASE=org.apache.log4j.jdbc.JDBCAppender   
log4j.appender.DATABASE.URL=jdbc:mysql://localhost:3306/test   
log4j.appender.DATABASE.driver=com.mysql.jdbc.Driver   
log4j.appender.DATABASE.user=root   
log4j.appender.DATABASE.password=   
log4j.appender.DATABASE.sql=INSERT INTO LOG4J (Message) VALUES ('[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n')   
log4j.appender.DATABASE.layout=org.apache.log4j.PatternLayout   
log4j.appender.DATABASE.layout.ConversionPattern=[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n  
log4j.appender.A1=org.apache.log4j.DailyRollingFileAppender   
log4j.appender.A1.File=SampleMessages.log4j   
log4j.appender.A1.DatePattern=yyyyMMdd-HH'.log4j'   
log4j.appender.A1.layout=org.apache.log4j.xml.XMLLayout  
###################   
#自定义Appender   
###################   
log4j.appender.im = net.cybercorlin.util.logger.appender.IMAppender  
log4j.appender.im.host = mail.cybercorlin.net   
log4j.appender.im.username = username   
log4j.appender.im.password = password   
log4j.appender.im.recipient = [corlin@yeqiangwei.com](mailto:corlin@yeqiangwei.com)  
log4j.appender.im.layout=org.apache.log4j.PatternLayout   
log4j.appender.im.layout.ConversionPattern =[framework] %d - %c -%-4r [%t] %-5p %c %x - %m%n  
三.高级使用  
实验目的：  
 1.把FATAL级错误写入2000NT日志  
 2. WARN，ERROR，FATAL级错误发送email通知管理员  
 3.其他级别的错误直接在后台输出  
实验步骤：  
 输出到2000NT日志  
 1.把Log4j压缩包里的NTEventLogAppender.dll拷到WINNT/SYSTEM32目录下  
 2.写配置文件log4j.properties  
# 在2000系统日志输出  
 log4j.logger.NTlog=FATAL, A8  
 # APPENDER A8  
 log4j.appender.A8=org.apache.log4j.nt.NTEventLogAppender  
 log4j.appender.A8.Source=JavaTest  
 log4j.appender.A8.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A8.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
3.调用代码：  
 Logger logger2 = Logger.getLogger("NTlog"); //要和配置文件中设置的名字相同  
 logger2.debug("debug!!!");  
 logger2.info("info!!!");  
 logger2.warn("warn!!!");  
 logger2.error("error!!!");  
 //只有这个错误才会写入2000日志  
 logger2.fatal("fatal!!!");  
发送email通知管理员：  
 1. 首先下载JavaMail和JAF,   
  <http://java.sun.com/j2ee/ja/javamail/index.html>  
  <http://java.sun.com/beans/glasgow/jaf.html>  
 在项目中引用mail.jar和activation.jar。  
 2. 写配置文件  
 # 将日志发送到email  
 log4j.logger.MailLog=WARN,A5  
 #  APPENDER A5  
 log4j.appender.A5=org.apache.log4j.net.SMTPAppender  
 log4j.appender.A5.BufferSize=5  
 [log4j.appender.A5.To=chunjie@yeqiangwei.com](mailto:log4j.appender.A5.To=chunjie@yeqiangwei.com)  
 [log4j.appender.A5.From=error@yeqiangwei.com](mailto:log4j.appender.A5.From=error@yeqiangwei.com)  
 log4j.appender.A5.Subject=ErrorLog  
 log4j.appender.A5.SMTPHost=smtp.263.net  
 log4j.appender.A5.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A5.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
 3.调用代码：  
 //把日志发送到mail  
 Logger logger3 = Logger.getLogger("MailLog");  
 logger3.warn("warn!!!");  
 logger3.error("error!!!");  
 logger3.fatal("fatal!!!");  
在后台输出所有类别的错误：  
 1. 写配置文件  
 # 在后台输出  
 log4j.logger.console=DEBUG, A1  
 # APPENDER A1  
 log4j.appender.A1=org.apache.log4j.ConsoleAppender  
 log4j.appender.A1.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A1.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
 2．调用代码  
 Logger logger1 = Logger.getLogger("console");  
 logger1.debug("debug!!!");  
 logger1.info("info!!!");  
 logger1.warn("warn!!!");  
 logger1.error("error!!!");  
 logger1.fatal("fatal!!!");  
--------------------------------------------------------------------  
 全部配置文件：log4j.properties  
 # 在后台输出  
 log4j.logger.console=DEBUG, A1  
 # APPENDER A1  
 log4j.appender.A1=org.apache.log4j.ConsoleAppender  
 log4j.appender.A1.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A1.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
# 在2000系统日志输出  
 log4j.logger.NTlog=FATAL, A8  
 # APPENDER A8  
 log4j.appender.A8=org.apache.log4j.nt.NTEventLogAppender  
 log4j.appender.A8.Source=JavaTest  
 log4j.appender.A8.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A8.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
# 将日志发送到email  
 log4j.logger.MailLog=WARN,A5  
 #  APPENDER A5  
 log4j.appender.A5=org.apache.log4j.net.SMTPAppender  
 log4j.appender.A5.BufferSize=5  
 [log4j.appender.A5.To=chunjie@yeqiangwei.com](mailto:log4j.appender.A5.To=chunjie@yeqiangwei.com)  
 [log4j.appender.A5.From=error@yeqiangwei.com](mailto:log4j.appender.A5.From=error@yeqiangwei.com)  
 log4j.appender.A5.Subject=ErrorLog  
 log4j.appender.A5.SMTPHost=smtp.263.net  
 log4j.appender.A5.layout=org.apache.log4j.PatternLayout  
 log4j.appender.A5.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n  
全部代码：Log4jTest.java

**[java]** [view plaincopyprint?](http://blog.csdn.net/zhshulin/article/details/37937365)

1. import org.apache.log4j.\*;
2. //import org.apache.log4j.nt.\*;
3. //import org.apache.log4j.net.\*;
4. /\*\*
5. \* @author yanxu
6. \*/
7. public class Log4jTest
8. {
9. public static void main(String args[])
10. {
11. PropertyConfigurator.configure("log4j.properties");
12. //在后台输出
13. Logger logger1 = Logger.getLogger("console");
14. logger1.debug("debug!!!");
15. logger1.info("info!!!");
16. logger1.warn("warn!!!");
17. logger1.error("error!!!");
18. logger1.fatal("fatal!!!");
19. //在NT系统日志输出
20. Logger logger2 = Logger.getLogger("NTlog");
21. //NTEventLogAppender nla = new NTEventLogAppender();
22. logger2.debug("debug!!!");
23. logger2.info("info!!!");
24. logger2.warn("warn!!!");
25. logger2.error("error!!!");
26. //只有这个错误才会写入2000日志
27. logger2.fatal("fatal!!!");
28. //把日志发送到mail
29. Logger logger3 = Logger.getLogger("MailLog");
30. //SMTPAppender sa = new SMTPAppender();
31. logger3.warn("warn!!!");
32. logger3.error("error!!!");
33. logger3.fatal("fatal!!!");
34. }
35. }

import org.apache.log4j.\*;

//import org.apache.log4j.nt.\*;

//import org.apache.log4j.net.\*;

/\*\*

\* @author yanxu

\*/

public class Log4jTest

{

public static void main(String args[])

{

PropertyConfigurator.configure("log4j.properties");

//在后台输出

Logger logger1 = Logger.getLogger("console");

logger1.debug("debug!!!");

logger1.info("info!!!");

logger1.warn("warn!!!");

logger1.error("error!!!");

logger1.fatal("fatal!!!");

//在NT系统日志输出

Logger logger2 = Logger.getLogger("NTlog");

//NTEventLogAppender nla = new NTEventLogAppender();

logger2.debug("debug!!!");

logger2.info("info!!!");

logger2.warn("warn!!!");

logger2.error("error!!!");

//只有这个错误才会写入2000日志

logger2.fatal("fatal!!!");

//把日志发送到mail

Logger logger3 = Logger.getLogger("MailLog");

//SMTPAppender sa = new SMTPAppender();

logger3.warn("warn!!!");

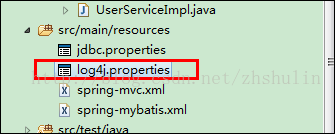
logger3.error("error!!!");

logger3.fatal("fatal!!!");

}

}

下面给出配置文件目录：



log4j.properties

#定义LOG输出级别

log4j.rootLogger=INFO,Console,File

#定义日志输出目的地为控制台

log4j.appender.Console=org.apache.log4j.ConsoleAppender

log4j.appender.Console.Target=System.out

#可以灵活地指定日志输出格式，下面一行是指定具体的格式

log4j.appender.Console.layout = org.apache.log4j.PatternLayout

log4j.appender.Console.layout.ConversionPattern=[%c] - %m%n

#文件大小到达指定尺寸的时候产生一个新的文件

log4j.appender.File = org.apache.log4j.RollingFileAppender

#指定输出目录

log4j.appender.File.File = logs/ssm.log

#定义文件最大大小

log4j.appender.File.MaxFileSize = 10MB

# 输出所以日志，如果换成DEBUG表示输出DEBUG以上级别日志

log4j.appender.File.Threshold = ALL

log4j.appender.File.layout = org.apache.log4j.PatternLayout

log4j.appender.File.layout.ConversionPattern =[%p] [%d{yyyy-MM-dd HH\:mm\:ss}][%c]%m%n

4.2.4、JUnit测试

      经过以上步骤（到4.2.2，log4j不配也没影响），我们已经完成了Spring和mybatis的整合，这样我们就可以编写一段测试代码来试试是否成功了。

4.2.4.1、创建测试用表

既然我们需要测试，那么我们就需要建立在[数据库](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F6%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%CA%FD%BE%DD%BF%E2&k0=%CA%FD%BE%DD%BF%E2&kdi0=8&k1=sql&kdi1=8&sid=b46ba3df2e460ce7&ch=0&tu=u1807396&jk=7547cd9e41b1a8bb&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)中建立一个测试表，这个表建的很简单，SQL语句为：

DROP TABLE IF EXISTS `user\_t`;

CREATE TABLE `user\_t` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(40) NOT NULL,

`password` varchar(255) NOT NULL,

`age` int(4) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

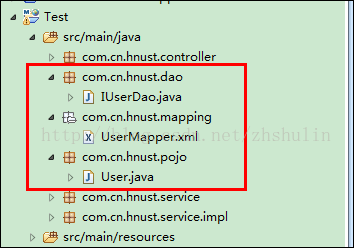
/\*Data for the table `user\_t` \*/

insert into `user\_t`(`id`,`user\_name`,`password`,`age`) values (1,'测试','sfasgfaf',24);

4.2.4.2、利用MyBatis Generator自动创建代码

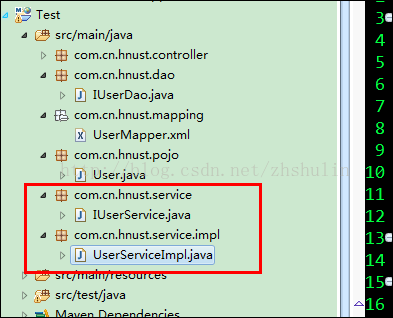
参考博文：<http://blog.csdn.net/zhshulin/article/details/23912615>

       这个可根据表自动创建实体类、MyBatis映射文件以及DAO接口，当然，我习惯将生成的接口名改为IUserDao，而不是直接用它生成的UserMapper。如果不想麻烦就可以不改。完成后将文件复制到工程中。如图：



4.2.4.3、建立Service接口和实现类

目录结构：



下面给出具体的内容：

IUserService.jave

package com.cn.hnust.service;

import com.cn.hnust.pojo.User;

public interface IUserService {

public User getUserById(int userId);

}

UserServiceImpl.java

package com.cn.hnust.service.impl;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

import com.cn.hnust.dao.IUserDao;

import com.cn.hnust.pojo.User;

import com.cn.hnust.service.IUserService;

@Service("userService")

public class UserServiceImpl implements IUserService {

@Resource

private IUserDao userDao;

@Override

public User getUserById(int userId) {

// TODO Auto-generated method stub

return this.userDao.selectByPrimaryKey(userId);

}

}

4.2.4.4、建立测试类

       测试类在src/test/[java](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=java&k0=java&k1=ajax&k2=xml&k3=%BF%F2%BC%DC&k4=%D7%E9%BC%FE&k5=%BF%D8%D6%C6%C6%F7&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)中建立，下面测试类中注释掉的部分是不使用Spring时，一般情况下的一种测试方法；如果使用了Spring那么就可以使用注解的方式来引入配置文件和类，然后再将service接口对象注入，就可以进行测试了。

       如果测试成功，表示Spring和Mybatis已经整合成功了。输出信息使用的是Log4j打印到控制台。

package org.zsl.testmybatis;

import [java](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=java&k0=java&k1=ajax&k2=xml&k3=%BF%F2%BC%DC&k4=%D7%E9%BC%FE&k5=%BF%D8%D6%C6%C6%F7&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)x.annotation.Resource;

import org.apache.log4j.Logger;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.test.context.ContextConfiguration;

import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.alibaba.fastjson.JSON;

import com.cn.hnust.pojo.User;

import com.cn.hnust.service.IUserService;

@RunWith(SpringJUnit4ClassRunner.class) //表示继承了SpringJUnit4ClassRunner类

@ContextConfiguration(locations = {"classpath:spring-mybatis.[xml](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=xml&k0=xml&k1=%BF%F2%BC%DC&k2=%D7%E9%BC%FE&k3=%BF%D8%D6%C6%C6%F7&k4=max&k5=spring&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)"})

public class TestMyBatis {

private static Logger logger = Logger.getLogger(TestMyBatis.class);

// private ApplicationContext ac = null;

@Resource

private IUserService userService = null;

// @Before

// public void before() {

// ac = new ClassPathXmlApplicationContext("applicationContext.xml");

// userService = (IUserService) ac.getBean("userService");

// }

@Test

public void test1() {

User user = userService.getUserById(1);

// System.out.println(user.getUserName());

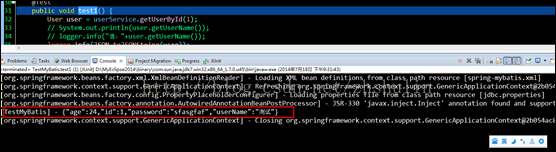
// logger.info("值："+user.getUserName());

logger.info(JSON.toJSONString(user));

}

}

测试结果：



至此，完成Spring和mybatis这两大[框架](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%F2%BC%DC&k0=%BF%F2%BC%DC&k1=%D7%E9%BC%FE&k2=%BF%D8%D6%C6%C6%F7&k3=max&k4=spring&k5=jsp&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=6&seller_id=1&di=128)的整合，下面在继续进行SpringMVC的整合。

4.3、整合SpringMVC

         上面已经完成了2大框架的整合，SpringMVC的配置文件单独放，然后在web.[xml](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=xml&k0=xml&k1=%BF%F2%BC%DC&k2=%D7%E9%BC%FE&k3=%BF%D8%D6%C6%C6%F7&k4=max&k5=spring&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)中配置整合。

4.3.1、配置spring-mvc.xml

配置里面的注释也很详细，在此就不说了，主要是自动扫描[控制器](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%D8%D6%C6%C6%F7&k0=%BF%D8%D6%C6%C6%F7&k1=max&k2=spring&k3=jsp&k4=java&k5=ajax&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=8&seller_id=1&di=128)，视图模式，注解的启动这三个。

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=jsp&k2=java&k3=ajax&k4=xml&k5=%BF%F2%BC%DC&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)framework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=jsp&k2=java&k3=ajax&k4=xml&k5=%BF%F2%BC%DC&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)framework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd">

<!-- 自动扫描该包，使SpringMVC认为包下用了@controller注解的类是[控制器](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%D8%D6%C6%C6%F7&k0=%BF%D8%D6%C6%C6%F7&k1=max&k2=spring&k3=jsp&k4=java&k5=ajax&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=8&seller_id=1&di=128) -->

<context:component-scan base-package="com.cn.hnust.controller" />

<!--避免IE执行AJAX时，返回JSON出现下载文件 -->

<bean id="mappingJacksonHttpMessageConverter"

class="org.[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=jsp&k2=java&k3=ajax&k4=xml&k5=%BF%F2%BC%DC&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)framework.http.converter.json.MappingJacksonHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>text/html;charset=UTF-8</value>

</list>

</property>

</bean>

<!-- 启动SpringMVC的注解功能，完成请求和注解POJO的映射 -->

<bean

class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">

<property name="messageConverters">

<list>

<ref bean="mappingJacksonHttpMessageConverter" /> <!-- JSON转换器 -->

</list>

</property>

</bean>

<!-- 定义跳转的文件的前后缀 ，视图模式配置-->

<bean class="org.[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=jsp&k2=java&k3=ajax&k4=xml&k5=%BF%F2%BC%DC&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)framework.web.servlet.view.InternalResourceViewResolver">

<!-- 这里的配置我的理解是自动给后面action的方法return的字符串加上前缀和后缀，变成一个 可用的url地址 -->

<property name="prefix" value="/WEB-INF/[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=jsp&k0=jsp&k1=java&k2=ajax&k3=xml&k4=%BF%F2%BC%DC&k5=%D7%E9%BC%FE&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=2&seller_id=1&di=128)/" />

<property name="suffix" value=".jsp" />

</bean>

<!-- 配置文件上传，如果没有使用文件上传可以不用配置，当然如果不配，那么配置文件中也不必引入上传[组件](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%D7%E9%BC%FE&k0=%D7%E9%BC%FE&k1=%BF%D8%D6%C6%C6%F7&k2=max&k3=spring&k4=jsp&k5=java&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=7&seller_id=1&di=128)包 -->

<bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<!-- 默认编码 -->

<property name="defaultEncoding" value="utf-8" />

<!-- 文件大小最大值 -->

<property name="[max](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F7%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=max&k0=max&k1=spring&k2=jsp&k3=java&k4=ajax&k5=xml&sid=8e3093d57e2cd2bf&ch=0&tu=u1807396&jk=1faef1398025f182&cf=29&fv=15&stid=9&urlid=0&luki=9&seller_id=1&di=128)UploadSize" value="10485760000" />

<!-- 内存中的最大值 -->

<property name="maxInMemorySize" value="40960" />

</bean>

</beans>

4.3.2、配置web.xml文件

         这里面对spring-mybatis.xml的引入以及配置的spring-mvc的Servlet就是为了完成SSM整合，之前2[框架](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%F2%BC%DC&k0=%BF%F2%BC%DC&k1=spring&k2=javaee&k3=xml&k4=%B3%CC%D0%F2%D4%B1&k5=jsp&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=8&seller_id=1&di=128)整合不需要在此处进行任何配置。配置一样有详细注释，不多解释了。

[web](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=web&k0=web&k1=%BF%F2%BC%DC&k2=spring&k3=javaee&k4=xml&k5=%B3%CC%D0%F2%D4%B1&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=7&seller_id=1&di=128).[xml](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=xml&k0=xml&k1=%B3%CC%D0%F2%D4%B1&k2=jsp&k3=struts&k4=web&k5=%BF%F2%BC%DC&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://java.sun.com/xml/ns/[javaee](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=javaee&k0=javaee&k1=xml&k2=%B3%CC%D0%F2%D4%B1&k3=jsp&k4=struts&k5=web&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=2&seller_id=1&di=128)"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0">

<display-name>Archetype Created Web Application</display-name>

<!-- Spring和mybatis的配置文件 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=javaee&k2=xml&k3=%B3%CC%D0%F2%D4%B1&k4=jsp&k5=struts&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)-mybatis.[xml](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=xml&k0=xml&k1=%B3%CC%D0%F2%D4%B1&k2=jsp&k3=struts&k4=web&k5=%BF%F2%BC%DC&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)</param-value>

</context-param>

<!-- 编码过滤器 -->

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.[web](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=web&k0=web&k1=%BF%F2%BC%DC&k2=spring&k3=javaee&k4=xml&k5=%B3%CC%D0%F2%D4%B1&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=7&seller_id=1&di=128" \t "_blank).filter.CharacterEncodingFilter</filter-class>

<async-supported>true</async-supported>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Spring监听器 -->

<listener>

<listener-class>org.[spring](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=spring&k0=spring&k1=javaee&k2=xml&k3=%B3%CC%D0%F2%D4%B1&k4=jsp&k5=struts&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128" \t "_blank)framework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 防止Spring内存溢出监听器 -->

<listener>

<listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>

</listener>

<!-- Spring MVC servlet -->

<servlet>

<servlet-name>SpringMVC</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

<async-supported>true</async-supported>

</servlet>

<servlet-mapping>

<servlet-name>SpringMVC</servlet-name>

<!-- 此处可以可以配置成\*.do，对应[struts](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=struts&k0=struts&k1=web&k2=%BF%F2%BC%DC&k3=spring&k4=javaee&k5=xml&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=6&seller_id=1&di=128)的后缀习惯 -->

<url-pattern>/</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>/index.[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=jsp&k0=jsp&k1=struts&k2=web&k3=%BF%F2%BC%DC&k4=spring&k5=javaee&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128" \t "_blank)</welcome-file>

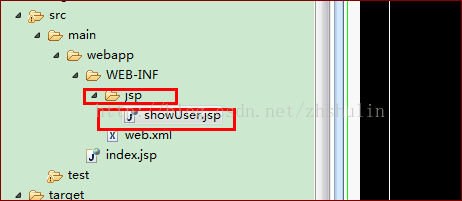
</welcome-file-list>

</web-app>

4.3.3、测试

        至此已经完成了SSM三大[框架](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%F2%BC%DC&k0=%BF%F2%BC%DC&k1=spring&k2=javaee&k3=xml&k4=%B3%CC%D0%F2%D4%B1&k5=jsp&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=8&seller_id=1&di=128)的整合了，接下来测试一下，如果成功了，那么恭喜你，如果失败了，继续调试吧，作为[程序员](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%B3%CC%D0%F2%D4%B1&k0=%B3%CC%D0%F2%D4%B1&k1=jsp&k2=struts&k3=web&k4=%BF%F2%BC%DC&k5=spring&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=4&seller_id=1&di=128)就是不停的与BUG做斗争！

4.3.3.1、新建jsp页面



showUser.[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F8%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=jsp&k0=jsp&k1=struts&k2=web&k3=%BF%F2%BC%DC&k4=spring&k5=javaee&sid=fd9ba18a51943992&ch=0&tu=u1807396&jk=3e11be6305f83de5&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)   此页面仅输出一下用户名，完成一个完整的简单流程。

<%@ page language="java" import="java.util.\*" pageEncoding="utf-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<[head](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=head&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=3&seller_id=1&di=128)>

<title>测试</title>

</head>

<body>

${user.userName}

</body>

</html>

4.3.3.2、建立UserController类

UserController.[java](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=java&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=1&seller_id=1&di=128)  [控制器](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%D8%D6%C6%C6%F7&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=5&seller_id=1&di=128)

package com.cn.hnust.controller;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import org.springframework.stereo[type](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=type&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=6&seller_id=1&di=128).Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import com.cn.hnust.pojo.User;

import com.cn.hnust.service.IUserService;

@Controller

@RequestMapping("/user")

public class UserController {

@Resource

private IUserService userService;

@RequestMapping("/showUser")

public String toIndex(HttpServletRequest request,Model [model](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=model&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=4&seller_id=1&di=128)){

int userId = Integer.parseInt(request.getParameter("id"));

User user = this.userService.getUserById(userId);

model.addAttribute("user", user);

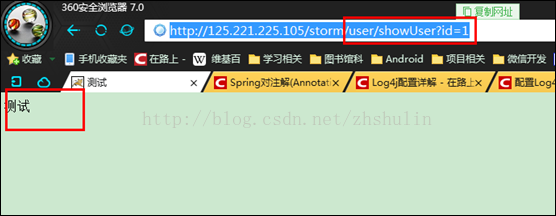
return "showUser";

}

}

4.3.3.3、部署项目

输入地址：localhost:8080/项目名称/user/showUser?id=1



至此，SSM三大[框架](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fwww%2E350351%2Ecom%2Fbianchengyuyan%2FSpring%2F318424%5F9%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=83045059_cpr&k=%BF%F2%BC%DC&k0=java&kdi0=8&k1=%BF%F2%BC%DC&kdi1=8&k2=head&kdi2=8&k3=model&kdi3=8&k4=%BF%D8%D6%C6%C6%F7&kdi4=1&k5=type&kdi5=8&sid=913263c290b4784a&ch=0&tu=u1807396&jk=cda4120eece758c4&cf=29&fv=15&stid=9&urlid=0&luki=2&seller_id=1&di=128)的整合就完成了，在此基础上可再添加其他功能。