IntelliJ IDEA 2016.2.1建立maven的ssm项目简单开发流程

Maven是基于项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的软件项目管理工具。

**准备：**

先装好IDEA（License

Server输入http://www.iteblog.com/idea/key，即可连网激活，以后使用中如果是断网情况下打开IDEA可能会需要再重新用该方法联网激活）和最新版的maven（最新3.5版的下载地址：https://maven.apache.org/download.cgi），maven的安装配置和java的配置差不多（安装配置看http://codingxiaxw.cn/2016/11/24/51-first-maven-project/），然后看这个链接IDEA用maven创建web项目http://blog.csdn.net/mercyT/article/details/51286825   
，建立web项目时选择自己安装的3.5的maven，不要用他默认自带的maven，然后就配置Tomcat，运行出现helloworld界面，web项目运行成功，就初步了解maven下搭建的web项目或后台项目了

IDEA下用maven搭建ssm框架的后台的详细步骤请看https://github.com/MovieDate/MovieDateServer.git项目中的doc包下的开发文档，如果没有时间看详细的步骤就看下面的0基础开发后台

**数据传输处理路径：**

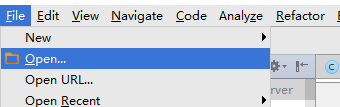
**前端通过URL调用控制层：controller，控制层调用业务层：service，业务层调用数据访问层：dao**

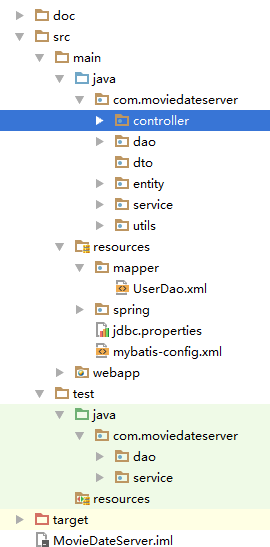
**0基础编写后台的步骤：主要是红色字体**

我们的约影吧后台代码：<https://github.com/MovieDate/MovieDateServer.git>

我已经写好了user\_tab表，剩下的表你们参照User来写

导入后台项目过程：选中项目即可以，如果报错再具体查看





目录解析和目录下主要放置什么文件：

doc：这里我放了一个后台项目的详细搭建步骤，如果有空可以看一看

main：后台项目文件

test：测试main里面dao和service里面的方法，放置测试类

entity:实体类

dao：数据访问对象的类即dao层类

service：业务层类

controller：控制层类

dto:传输类，例如约影功能需要的实体类，里面定义的变量有对应user\_tab表的用户的姓名，性别等和约影表中的电影名，地名，约影详情等。

utils：工具包，如时间格式的转换工具类TimeUtil

resources：一些配置的资源文件

mapper：对应dao包中的类，实现访问数据库操作的SQL语句

spring：mybatis，spring，springMVC的整合配置文件

jdbc.properties:连接数据库文件

mybatis-config.xml:mybatis配置文件

Pom.xml;POM包括了所有的项目信息，jar包的依赖等

groupId:项目或者组织的唯一标志，并且配置时生成路径也是由此生成，如 org.myproject.mojo生成的相对路径为：/org/myproject/mojo

artifactId:项目的通用名称

version:项目的版本

为项目导入jar包的命令：

Mvn install:install-file -Dfile=D:\apache-maven-3.5.0workspace\repo\commons-io-2.5.jar -DgroupId=com.alibaba.myfastjson -DartifactId=myfastjson -Dversion=1.2.7 -Dpackaging=jar （每一项用空格隔开，是写在同一行的，配置好maven后，在cmd窗口执行）

同时在pom.xml中添加该jar包的依赖

*<!--自定义json转换-->*<**dependency**>  
 <**groupId**>com.alibaba.myfastjson</**groupId**>  
 <**artifactId**>myfastjson</**artifactId**>  
 <**version**>1.2.7</**version**>  
</**dependency**>

这样就可以使用该jar包中的接口，方法

1. 根据数据库表在entity包下建立相应的实体类，里面的变量对应表中的字段，例如项目中的User.java

数据访问层

1. 在dao包建立与entity包下实体类相对应的\*Dao.java文件，例如UserDao.java，这里定义一些通过参数进行简单增删改查的方法，只需要定义，实现在对应的\*Dao.xml里实现，\*Dao.xml文件需要在resources/mapper下创建，\*Dao.xml中主要是SQL语句对数据库进行增删改查，这就是mybatis做的工作，属于dao层。\*Dao.xml中的方法就能够增删改查数据库里的数据了，但一般\*Dao.xml里只是简单的增删改查，没有复杂的逻辑数据处理，数据的处理放在service层，即\*Service.xml里面。对dao里面的方法进行测试，见test包的dao包中的测试类，测试是为了提前找出问题

业务层：可以通过注解方式调用数据访问层中的所有方法，即dao中的方法

1. 在service包创建与dao包中相对应的\*Service.java文件，在\*Service.java里定义一些整合\*Dao.java中整合增删改查的方法，在impl下对应的\*ServiceImpl.java里实现这些定义的方法，例如可以注入别的表如post\_tab表对应的PostDao，这样就可以根据需求整合多表的增删改查。对service里面的方法进行测试，见test包中的service包中的测试类

控制层：可以通过注解方式调用业务层中的所有方法，即service中的方法

1. 在controller包创建\*Controller.java，根据前端需要获取什么数据建立相应的方法来为前端提供数据。方法里面对从service里获取来的数据进行进一步的加工处理，为前端提供他所需要的数据或者信息。

**例如登录功能，前端发来phone和password这两个请求参数，我们的登录方法里就获取这两个传过来的参数，通过service和dao对数据的处理，能否在数据库查找到对应的phone和password的用户，再返回给前端 验证账号密码是否正确的信息，前端就可以做进一步的操作处理**

测试controller中的方法：

火狐浏览器安装HttpRequest插件

**//tomcat配置好的项目地址是 + controller配好的RequestMapping + controller里面的接口方法配置好的RequestMapping  
// 就是网络请求时的地址，即访问controller中某方法的网络地址URL  
// 例如：http://localhost:8080/userController/allUser --》就访问了findAllUserDESC（）方法，APP那边也是访问这样的链接来访问后台，请求数据和信息**

