搭建SSH项目注意：

# 1 SSH中各个配置文件的意义

1. web.xml：用来配置spring监听器，设置监听，加载上下文；

2. applicationContext.xml：为数据源注入session工厂，配置一些action里面的常类等的属性；

3. hibernate.cfg.xml：hibernate对jdbc进行了简单的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库，这里设定了数据库连接的信息；

4. struts.xml：struts的核心配置文件，负责管理应用中的action映射以及该action包含的result定义以及处理结果返回的页面等；

5. \*.hbm.xml：为各个bean的描述文件。

摘抄来自：

<https://blog.csdn.net/holmes_conan/article/details/41154073>

<https://blog.csdn.net/sinat_24048051/article/details/68951012>（这个搭建过程好像不错）

# SSH连接数据库导入实体类时SessionFactory创建不了

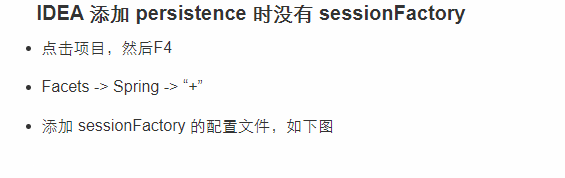
方法一：（不是很实用，Persistence视图多出一个JPA）Persistence视图的实体映射过程，报错“Please choose persistence unit”

解决：先到项目结构中的Modules中添加一个JPA然后再去操作.

解决来自：

<https://blog.csdn.net/handsome2013/article/details/81638514>

方法二：（如图，忘了收藏网址，觉得这个才是原因）：

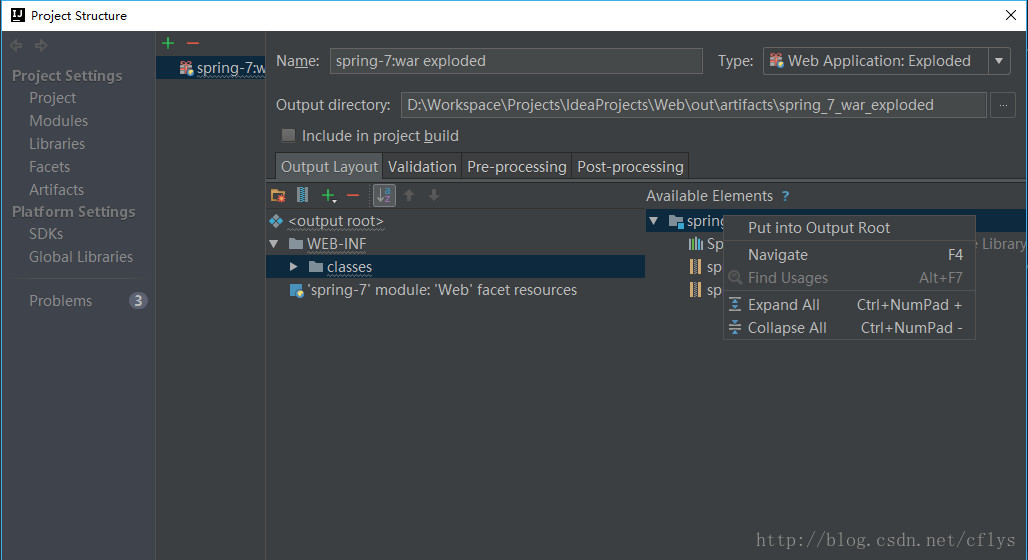


# Tomcat启动报错如下：

# org.apache.catalina.core.StandardContext.listenerStart Error configuring application listener of class [org.springframework.web.context.ContextLoaderListener]

# ****java.lang.ClassNotFoundException: org.springframework.web.context.ContextLoaderListener****

解决方法：（注意选中自己的包）Ctrl+Alt+Shift+S弹出Project Structure窗口，在Output Layout标签中找到Available Elements，鼠标右击下面的元素，点击Put into Output Root，执行后，在WEB-INF会增加lib目录，里面是项目引用的jar包，点击OK。再次部署应用就能成功运行了。：



# 4 找不到Persistence

**操作Persistence，自动生成Entity；我们可以看到在Idea界面里面看不到Persistence选项（将鼠标悬浮在左下角的正方形那里，截图如下），因为一开始使用maven来搭建项目，我们并没有显示引入hibernate，所以没有Persistence界面，所以我们作如下操作来显示加入Hibernate; 点击 File-----> Project Structure------>Modules----->点击+-------->选择Hibernate------>Apply------>OK;**

# 5 配置出c3p0，报错：

# Caused by: com.mchange.v2.resourcepool.CannotAcquireResourceException: A ResourcePool could not acquire a resource from its primary factory or source.

解决：（逃避式解决）同样的SSH项目重新搭建，c3p0报错以上错误，改了一天一夜，网上n多建议都无法解决，检查了很多遍，配置都没什么问题，不想吐槽了，换了一个连接池druid，一次成功，泪奔。（拉黑c3p0连接池，找个时间学习一下几个连接池的区别）

# 6 SSH各个文件和包的配置

## 6.1 Action类：

@Controller：控制类；

@Scope: 顾名思义就是作用域的意思，表示这个struts2的action的作用范围每一次调用的时候就会新建一个action，如果@Scope中的prototype换成singleton(也就是单例)就是会出现，每一次请求用同一个action；

Action类要继承ActionSupport；

@Resource:注入，一般在Action类是注入service的实例；

## 6.2 服务层：

@Service:放在service的实现类上注入服务；

@Resource:注入Dao层实例；

## 6.3 Dao层

@Transactional(rollbackFor = Exception.class):声明式事务管理，当作用于类上时，该类的所有 public 方法将都具有该类型的事务属性，同时，我们也可以在方法级别使用该标注来覆盖类级别的定义。

在项目中，@Transactional(rollbackFor=Exception.class)，如果类加了这个注解，那么这个类里面的方法抛出异常，就会回滚，数据库里面的数据也会回滚。

在@Transactional注解中如果不配置rollbackFor属性,那么事物只会在遇到RuntimeException的时候才会回滚,加上rollbackFor=Exception.class,可以让事物在遇到非运行时异常时也回滚；

***@Transactional注解的全部属性详解***

*@Transactional属性*

|  | | |
| --- | --- | --- |
| **属性** | **类型** | **描述** |
| value | String | 可选的限定描述符，指定使用的事务管理器 |
| propagation | enum: Propagation | 可选的事务传播行为设置 |
| isolation | enum: Isolation | 可选的事务隔离级别设置 |
| readOnly | boolean | 读写或只读事务，默认读写 |
| timeout | int (in seconds granularity) | 事务超时时间设置 |
| rollbackFor | Class对象数组，必须继承自Throwable | 导致事务回滚的异常类数组 |
| rollbackForClassName | 类名数组，必须继承自Throwable | 导致事务回滚的异常类名字数组 |
| noRollbackFor | Class对象数组，必须继承自Throwable | 不会导致事务回滚的异常类数组 |
| noRollbackForClassName | 类名数组，必须继承自Throwable | 不会导致事务回滚的异常类名字数组 |

（摘抄自：

<https://www.cnblogs.com/clwydjgs/p/9317849.html>）

@Repository("userDao") ：用于将数据访问层 (DAO 层 ) 的类标识为 Spring Bean，需要在 XML 配置文件中启用Bean 的自动扫描功能，这可以通过<context:component-scan/>实现如此，我们就不再需要在 XML 中显式使用 <bean/> 进行Bean 的配置；

@Component 是一个泛化的概念，仅仅表示一个组件 (Bean) ，可以作用在任何层次，另外，用户可以创建自定义的注解，然后在注解上标注 @Component，那么，该自定义注解便具有了与所@Component 相同的功能。不过这个功能并不常用；

（以上两个注解解释摘抄自：

<https://www.cnblogs.com/zhangzhifeng/p/4228498.html>）

# 7 log4j的配置

https://blog.csdn.net/PENGYAO\_O/article/details/78857095

# 8 表单提交

如果以form表单的post形式提交，action可以进行页面的跳转，如果使用ajax的形式进行跳转，则不会进行页面的跳转，返回302状态码，需要手动获得页面路径跳转。