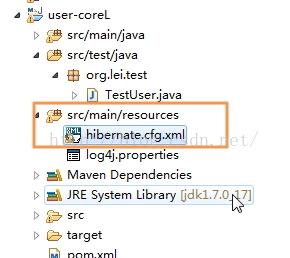
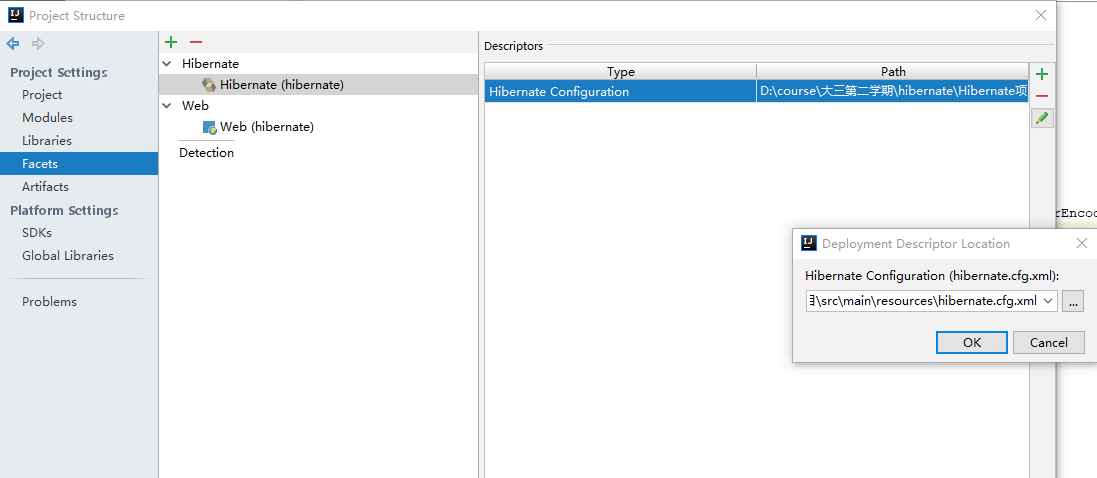
1. hibernate.cfg.xml和lo4j.properties的文件必须放在 src/main/resources目录下面。如下图所示：



1. 在pom.xml导入mysql依赖
2. 在pom.xml导入hibernate4.3.11依赖
3. 在 src/main/resources创建hibernate.cfg.xml文件.
4. *<?***xml version="1.0" encoding="UTF-8"***?>***<!DOCTYPE hibernate-configuration PUBLIC  
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd"*>***<**hibernate-configuration**>  
    <**session-factory**>  
    <**property name="connection.username"**>root</**property**>  
    <**property name="connection.password"**></**property**>  
    <**property name="connection.driver\_class"**>com.mysql.jdbc.Driver</**property**>  
    <**property name="connection.url"**>jdbc:mysql:///demo1?userUnicode=true**&amp;**characterEncoding=UTF-8</**property**>  
    <**property name="dialect"**>org.hibernate.dialect.MySQLDialect</**property**>  
     
    <**property name="c3p0.max\_size"**>20</**property**>  
    <**property name="c3p0.min\_size"**>1</**property**>  
    <**property name="c3p0.timeout"**>5000</**property**>  
    <**property name="c3p0.max\_statements"**>100</**property**>  
    <**property name="c3p0.acquire\_increment"**>2</**property**>  
    <**property name="c3p0.idle\_test\_period"**>3000</**property**>  
     
    <**property name="show\_sql"**>true</**property**>  
    <**property name="format\_sql"**>true</**property**>  
    <**property name="hbm2ddl.auto"**>update</**property**>  
     
    <**mapping class="domain.Student"**/>  
    </**session-factory**>  
   </**hibernate-configuration**>
5. 打开project Structure ,添加hibernate并配置其对应的配置文档：



1. 创建SessionFactory工厂：
2. **import** org.hibernate.cfg.Configuration;  
   **import** org.hibernate.service.ServiceRegistry;  
     
   **public class** MySessionFactory {  
    **private static** SessionFactory *sessionFactory* ;  
    **private static** Configuration *config*;  
    **static** {  
    */\*  
    Configuration类用来管理我们的配置文件的信息的，  
    通过它，我们可以通过创建一个configuration实例来管理相应的配置文档  
    \*/  
    config* = **new** Configuration().configure();  
    }  
    **public static** SessionFactory getSessionFactory(){  
    **if** (*sessionFactory* == **null**){  
     
    *//创建服务注册对象  
    //实例化一个标准化服务注册创建者* StandardServiceRegistryBuilder serviceRegistryBuilder = **new** StandardServiceRegistryBuilder();  
    *//1.调用标准化服务注册创建者的applySettings方法调用配置文件内容  
    // 2.再用build（）方法创建一个服务注册对象  
    //applySettings方法的参数是配置文件的内容[config.getProperties()]* ServiceRegistry serviceRegistry = serviceRegistryBuilder.applySettings(*config*.getProperties()).build();  
    *//创建会话工厂  
    sessionFactory* = *config*.buildSessionFactory(serviceRegistry);  
    */\*  
    等同于上面的三句  
    sessionFactory= config.buildSessionFactory(new StandardServiceRegistryBuilder().  
    applySettings(config.getProperties()).build());  
    \*/* **return** *sessionFactory*;  
    }  
    **return** *sessionFactory*;  
    }  
    **public static void** close(){  
    **if** (*sessionFactory* !=**null**){  
    *sessionFactory*.close();  
    }  
    }  
   }