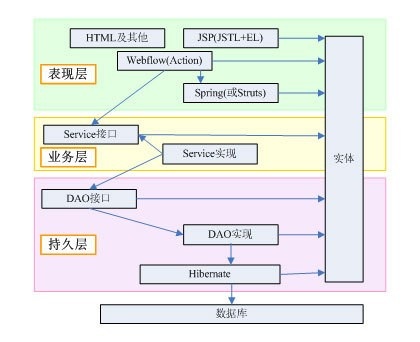
01-Hibernate的概述.avi

A：java ee 三层结构模型



三层架构关注的是表现层的怎么显示以及如何控制，业务层关注的是数据逻辑，持久层关注的是如何操作数据；

B：hibernate简单介绍

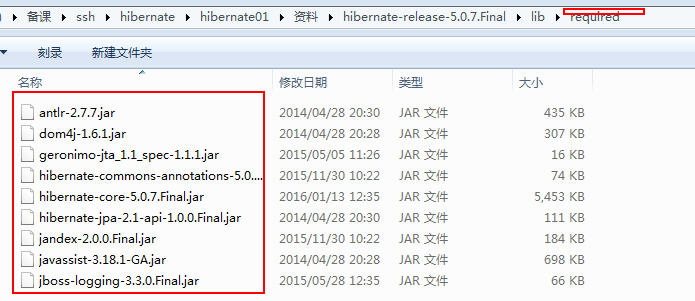
hibernate框架处于java ee三层架构中的持久层，该框架属于ORM（Object Relational Mapping） 框架一种，处理java 对象和数据库表之间的映射。

注意：认识javaee三层架构模型，以及指导hibernate是orm框架。

02-Hibernate入门：Hibernate的环境搭建.avi

A：本地搭建hibernate环境

下载地址：<https://sourceforge.net/projects/hibernate/files/hibernate-orm/>

1. 从下载好的hibernate框架中找到lib文件夹，然后选择required，将该文件夹下所有jar以及数据mysql的驱动jar、日志jar拷贝到工程目录下；  
    



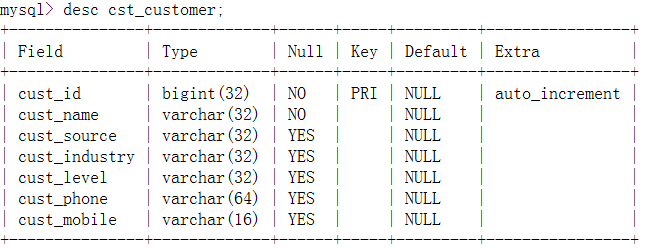


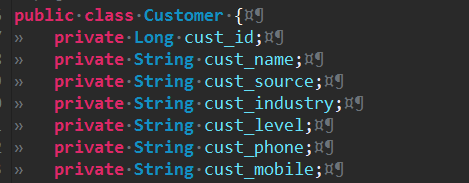
1. 在mysql数据库中创建数据库以及表





3． 创建于表对应的实体bean





注意：加入jar包，创建数据库表以及实体，为后面hibernate映射做铺垫；

03-Hibernate入门：Hibernate的映射文件的创建.avi

A: 新建一个hibernate中实体与表之间的映射文件  
 一般在实体类下面新建一个xml文件。取名一般遵循着类名.hbm.xml

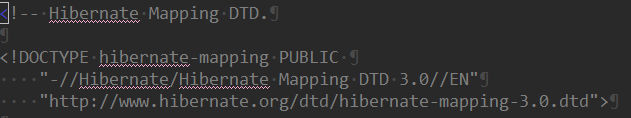


B：hibernate中实体与表之间的映射文件配置

1. 引入约束

 下面有

 约束，打开将



将约束拷贝到我们的hibernate映射文件中。

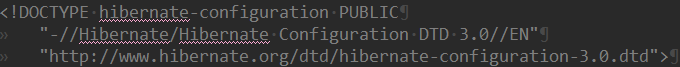
注意：实体类写好了之后将要新建一个映射文件，取名遵循类名.hbm.xml。然后再到该映射文件中引入约束；

04-Hibernate入门：Hibernate的核心配置文件.avi

A：新建hibernate的核心配置文件用于连接数据库

1. 配置hibernate.cfg.xml文件的约束头文件

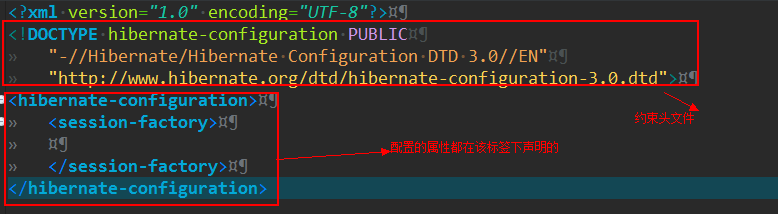
一般在src下面新建一个hibernate.cfg.xml文件。在中找到 dtd约束文件。在该约束文件中找到xml文件头



复制拷贝到hibernate.cfg.xml文件中。

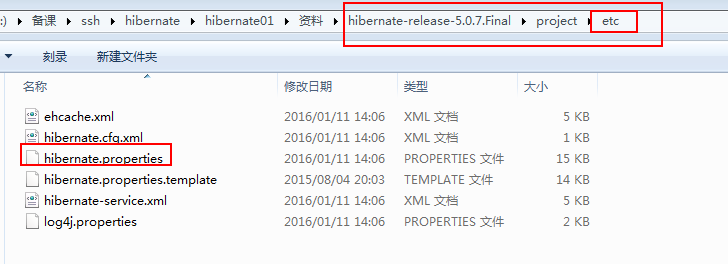
1. 配置hibernate.cfg.xml中的属性

hibernate.cfg.xml文件属性配置都是在如下标签中声明的：



1. hibernate.cfg.xml要配置的属性

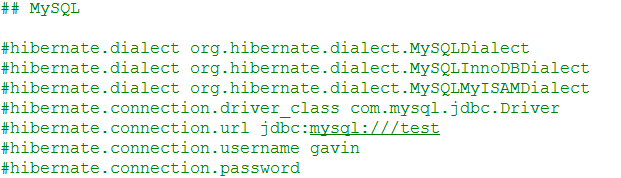
找到下载包中的project文件夹，找到etc文件夹，hibernate.cfg.xml有关的属性声明都在该文件中有说明。



该properties文件中涉及属性分类为如下标记：



如上图，该分类就是hibernate关联的不同的数据库，找到mysql



左边的是属性，右边的是属性值

更多的属性通过文件知道该标记。当然hibernate.cfg.xml文件中属性配置一般都会 涉及到平台数据库连接、连接池、引入映射文件等。

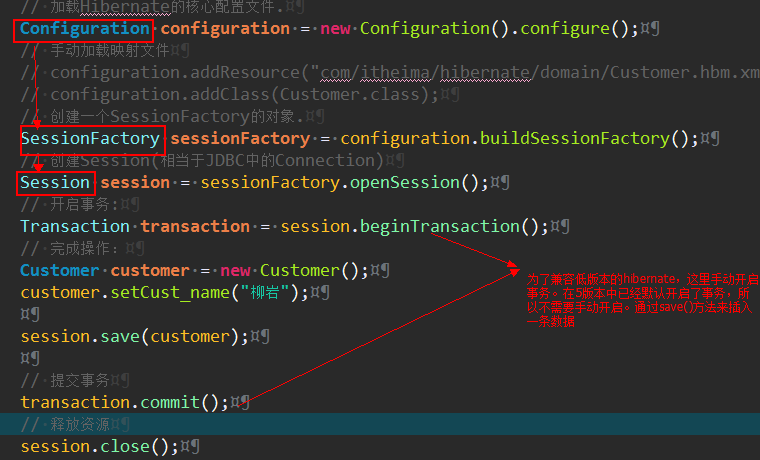
注意：hibernate中的所有属性都是在<hibernate-configuration><session-fatory><property></property><mapping/></session-fatory></hibernate-configuration>

05-Hibernate入门：Hibernate的测试代码编写.avi

A：hibernate测试插入一条数据

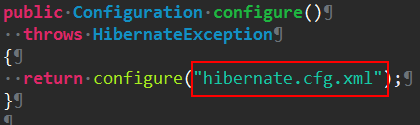
hibernate测试代码在junit中来完成的。

hibernate数据库的操作，需要拿到session,该session就相当于connection。



B：Session操作数据库方法

Configuration通过configure()来获取Configuration对象。该方法根据参数有很多种。如果默认不写，就直接加载hibernate.cfg.xml文件



该测试案例中只介绍了save()用来插入一条数据。后面还会介绍update(),delete(),get()方法来操作修改、删除、查询操作。

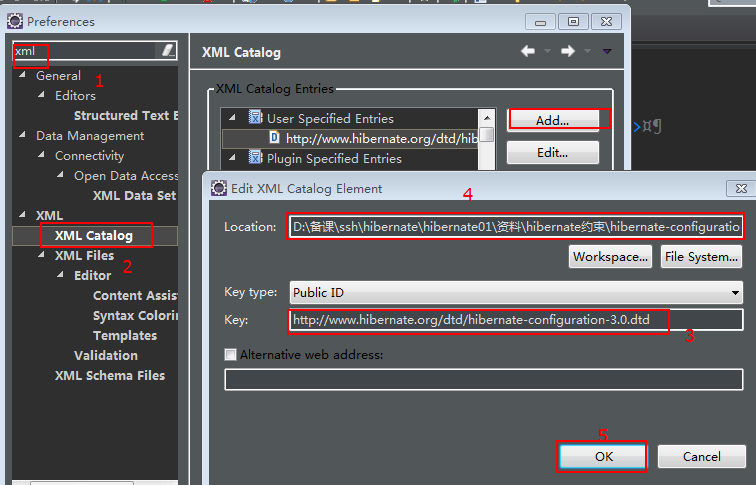
注意：Configuraion 得到SessionFactory，SessionFactory得到Session。通过session操作数据库。

06-Hibernate常见配置：配置文件的提示.avi

A：在没有联网的情况，配置本地约束解决提示

复制dtd的网络地址：<http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd>

选择下图中的Key,粘贴进去。然后点击File System找到本地dtd约束文件。按照序号操 作完成配置本地约束。

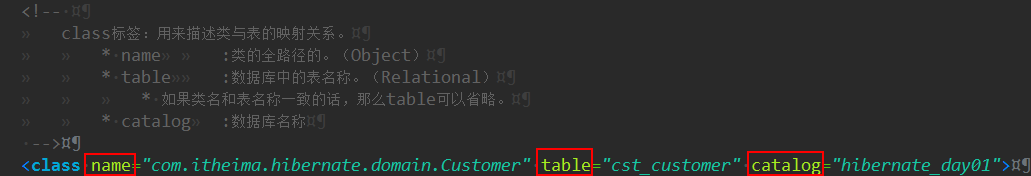


注意：以上解决的是本地没有提示，注意要首先复制网络dtd地址。

07-Hibernate常见配置：映射文件的配置.avi

A: hibernate实体表映射文件详细介绍

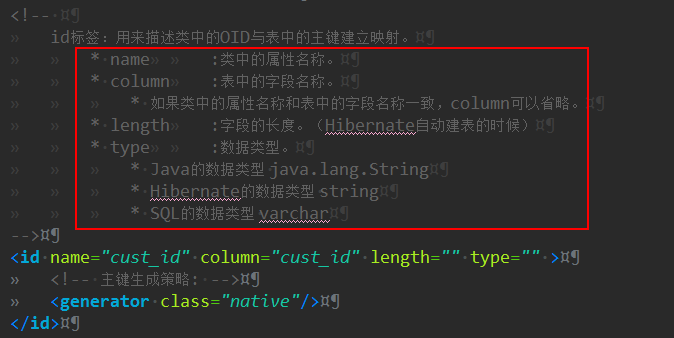
1. 实体类与表<class></class>



catalog代表数据库，可以省略。

1. 主键<id></id>

如果不在数据库中创建表，hibernate可以根据标签在数据库总创建表；



1. 实体属性与表列名<property></property>

name和column对应，如果实体类属性名和表列名相同，则可以省略column

属性。

注意：一般都会首先创建数据库和表，所以都会存在实体类属性和表列名之间的映射。

08-Hibernate常见配置：核心配置文件的配置.avi

A：properties文件配置hibernate核心属性

所有的的properties文件中属性配置格式都是key=value；找到hibernate-release-5.0.7.Final/project/etc/hibernate.properties；将该属性前面的“#”去掉，将空格改成“=”；

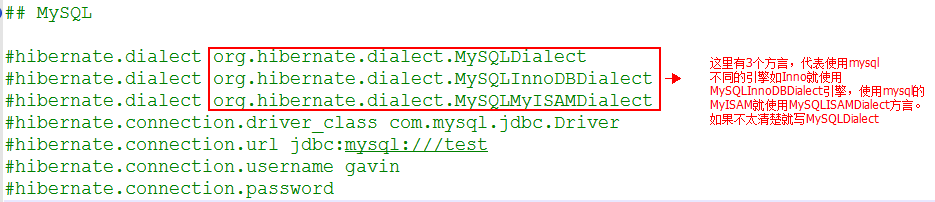


以properties文件配置hibernate核心属性，存在如下缺点：

1. 无法加载hibernate的映射文件，需要手动加载
2. 结构不清晰

B：xml文件配置hibernate核心属性

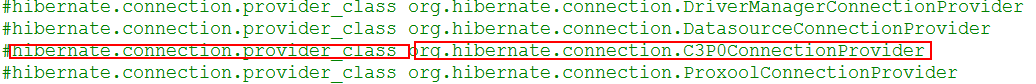
1. 数据库配置



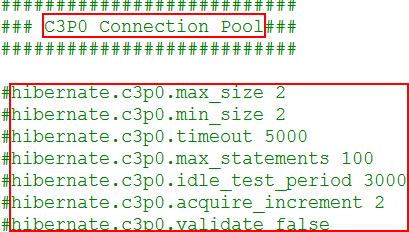
1. 连接池配置

如果选择c3p0作为连接池。这个时候就得提供“Plugin ConnectionProvider”。选择C3P0ConnectionProvider





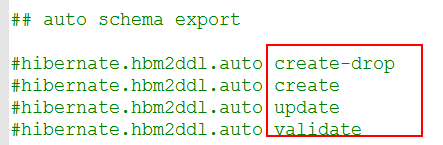
至于C3P0连接池属性的配置，找到



1. 其他各项配置(显示sql语句，格式等方面)

找到。选择合适的属性配置即可。需要注意的hibernate.hbm2ddl.auto属性的值有多个，hbm2ddl官方给出的解释是：





none: 不自动创建数据库表

create-drop: 如果该表存在就将原表删除，然后再创建新的表，sessionFactory.close()才能最终将该表drop;不执行sessionFactory.close()，则不会最后drop;

create:每次都会先删除原来的表然后根据实体映射文件创建新的表

update:如果数据库中有表使用原来的表。如果没有表会创建一个表。而且可以更新原有表结构。

validate:不会创建表，会校验表和数据库

1. 加载映射文件(<mapping></mapping>标签来完成的)



09-Hibernate常用API：Configuration对象.avi

A：Configuration对象

1. 加载核心配置文件

创建Configuration对象： Configuration config=new Configuraion();这样默认的是去寻找hibernate.properties文件

Configuration config=new Configuration().configure();默认去加载hibernate.cfg.xml文件。

1. 加载映射文件

如果不在hibernate.cfg.xml文件中通过<mapping/>来完成加载映射文件，则需要手动加载xxx.hbm.xml文件。

config.addResource(“com/itheima/hibernate/domain/Customer.hbm.xml”);

config.addClass(Customer.class);来完成加载映射文件

注意：熟悉Configuraion对象的创建，默认是回去加载hibernate.cfg.xml文件，所以在创建的时候一般默认取该名字，且放在src下面；

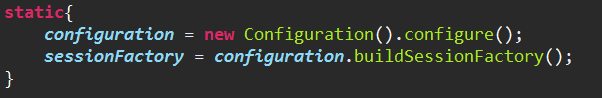
10-Hibernate常用API：SessionFactory抽取工具类.avi

A：SessionFactory

SessionFactory对象用于管理Session,连接池,管理二级缓存等。该对象是一个线程安全对象，一般一个项目只有一个该对象；

SessionFactory sessionFactory=config.buildSessionFactory();

该对象的创建可以放在一个工具类中，放在Static代码块中。



而Session对象是就是通过SessionFactory对象来创建的

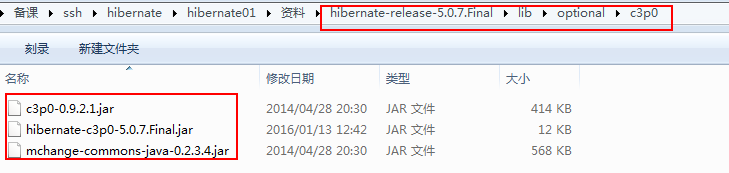
Session session=SessionFactory.openSession();

注意：SessionFactory对象是通过Configuraion对象创建的，并该对象是线程安全的。

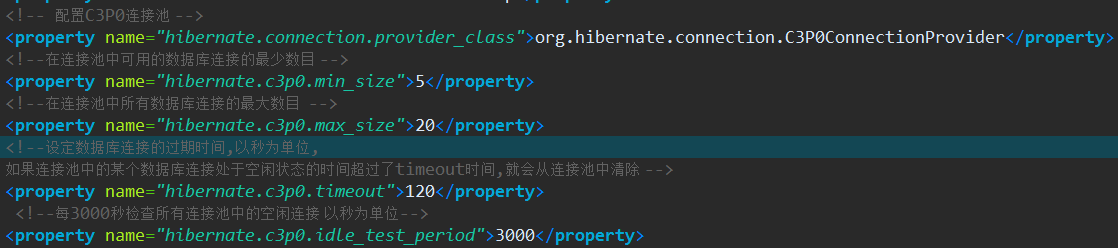
11-Hibernate常用API：SessionFactory配置c3p0.avi

A：hibernate完成c3p0连接池的配置

1. 导入c3p0的相关的jar(/lib/optional/c3p0/)



1. c3p0相关属性在hibernate.cfg.xml中配置（不要记，需要学会如何去找）



注意：c3p0连接池配置去project/etc/hibernate.properties中文件找。

12-Hibernate常用API：Session的get和load方法.avi

A：Session对数据库进行CRUD操作

Session对象是非线程安全对象，是单线程对象，作为操作数据库的桥梁，内部维护了hibernate的一级缓存。

1. 保存对象(save)

session.save(customer)

1. 查询(get,load)





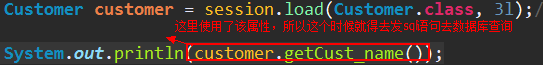
get和load区别：

a: get立即发送sql去查询，load是在commit()操作的时候才会去发送sql语句去执行。有一种情况若是在使用该对象其他非主键属性值的时候会发送sql语句。

b: 同时查询到返回的对象也不同，get直接是返回该实体对象，而load返

回的该实体对象的代理对象。该代理对象其实就是实体对象的子类。

c: 当查询为空的时候，get查询返回的是null,而load返回的是ObjectNotFoundException

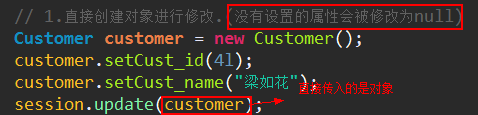


注意：Session对象的get()方法和load()方法都可以用来查询数据。两者之间的区别是load这里用到了一个延迟设计，直到必须用到sql语句的时候才会去发送sql语句；并且load方法返回到对象还跟get不一样，返回的直接是实体类的子类代理对象。

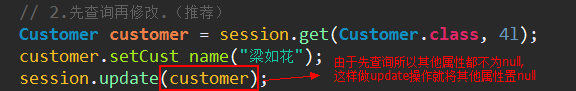
13-Hibernate常用API：Session的update和delete方法.avi

A：Session对象的update(实体对象)修改数据

1. 直接传入实体对象去修改

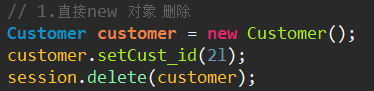


1. 先查询再去修改（推荐）



B：Session对象的delete(实体对象)删除数据

1. 直接传入实体对象去删除



1. 先查询再去删除数据（推荐）



这样先查询再删除，为后面做级联操作做铺垫。推荐使用这种方式；

注意：Session对象的update()，delete()方法操作数据，传入的参数都是一个实体对象，底层根据该对象进行拼接sql语句。需要注意的是update()要修改什么属性就只改变该属性，这时候需要先查询再修改。delete()根据你传入的对象的参数来进行删除数据。

14-Hibernate常用API：Transaction对象.avi

A：Transaction对象

hibernate默认并没有开启事务，所谓没有默认开启，就是你需要手动写start transaction 然后commit才能真正地操作数据库。

Transaction transaction=session.beginTrasaction();

transaction.commit();提交事务

transaction.rollback();回滚事务

注意：事务Transaction对象是从session中获取的，常用的是提交事务commit()和回滚事务rollback();

15-今天内容总结.avi

A：hibernate概述

持久层框架

下载hibernate框架，对hibernate下载框架包结构熟悉，知道每一个文件夹的作用；

B：hibernate应用的步骤

1. 导入jar(lib/required ,lib/optional)
2. 实体类和xx.hbm.xml映射文件

dtd约束文件，

<hibernate-mapping>

<class name=”” table=””>

<id name=”” column=””>

<generator class=””></generator>

</id>

<property name=”” column=””></property>

</class>

</hibernate-mapping>

1. hibernate.cfg.xml核心配置文件

找到/project/etc/hibernate.properties文件

<hibernate-configuraion>

<session-factory>

<property name=””></property>

<mapping resource=”” / >

</session-factory>

</hibernate-configuraion>

C：hibernate的常用的api

Configuration configuration=new Configuration().configure();

SessionFactory sessionFactory=configuration.buildSessionFactory();

Session session=sessionFactory.openSession();

session.update(实体对象);

session.delete(实体对象);

session.get();

session.load();

session.save(实体对象);

Transaction transaction=session.beginTransaction();

session.commit();

session.rollback();