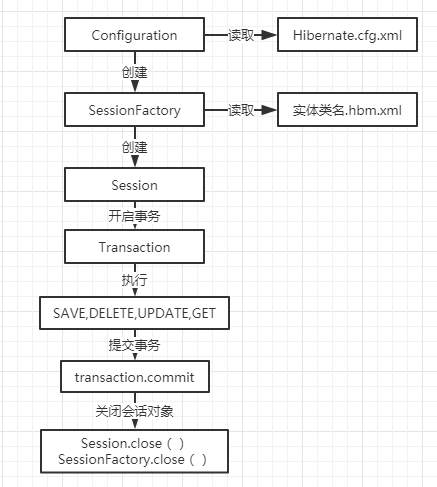
1. Hibernate基本概念
2. 定义
3. 搭建开发环境
4. 工作流程
5. session对象详解
6. 单表操作
7. 单一主键生成策略
8. hibernate中类型和数据库中类型
9. 对象类型
10. 组件属性
11. 单表的增删改查
12. 关联映射
13. 一对多
14. 多对一
15. Hibernate基础概念
16. 定义
17. Hibernate是一个开源的ORM框架，ORM(Object/Relationship Mapping):对象/关系映射。
18. Hibernate充当一个持久层，把程序中的实体类，持久化到数据库的表中。
19. 搭建Hibernate开发环境 -- 具体见其他文档。
20. Hibernate工作流程



1. Configuration对象：配置文件对象，读取hibernate配置文件hibernate.cfg.xml
2. SessionFactory对象：读取对象/关系映射文件 xxx.hbm.xml
3. session对象：数据库链接对象，可以理解为操作数据库的对象。

session和connection是多对一的关系，一个connection可以供多个session使用。如果要操作数据库，必须通过session的方法来访问，且操作对象是数据库表映射的实体类。

相关方法：save() ，update()， delete()， createQuery()

1. Transaction:使用session操作数据库需要开启的事务

所有对数据库的操作必须先开启事务，再手工提交事务才会执行，可以设置事务回滚。

如果要改变为自动提交事务，可以调用session.doWork()方法获取connection,设置为自动提交事务模式。

4. Session对象详解

4.1 获取Session对象的方法：

(1) SessionFactory.openSession() -- 默认

(2) SessionFactory.getCurrentSession() -- 手工配置hibernate.cfg.xml文件

jdbc本地事务

<propertyname = “hibernate.current\_session\_context\_class”>thread</property>

jta全局事务

<propertyname = “hibernate.current\_session\_context\_class”>jta</property>

4.2 两种方法区别

(1) openSession()相当于多例子，getCurrentSession()相当于单例。

(2) openSession要手动关闭，否则不会自动关闭，再次创建会存在两个session，两者hashcode不同。

(3) getCurrentSession会在事务提交后自动关闭，再次创建的session与已关闭的session的hashcode是一致的。

二、Hibernate单表操作

1．单一主键生成策略：

(1) assigned 由java应用程序负责生成（手工赋值），通过java中赋值的。

(2) native 由底层数据库自动生成标识符，如果是MySQL就是increment，如果是Oracle就是sequence，等等。

2．hibernate中类型和数据库中类型





1. 对象类型



1. 组件属性

实体类中某个属性属于用户自定义类的对象，实体类中的内部对象。

1. 增删改查

(1) 单表的增删改查CRUD 对应 session中的方法

session.save()：insert语句

session.update()：update语句

session.delete()：delete语句

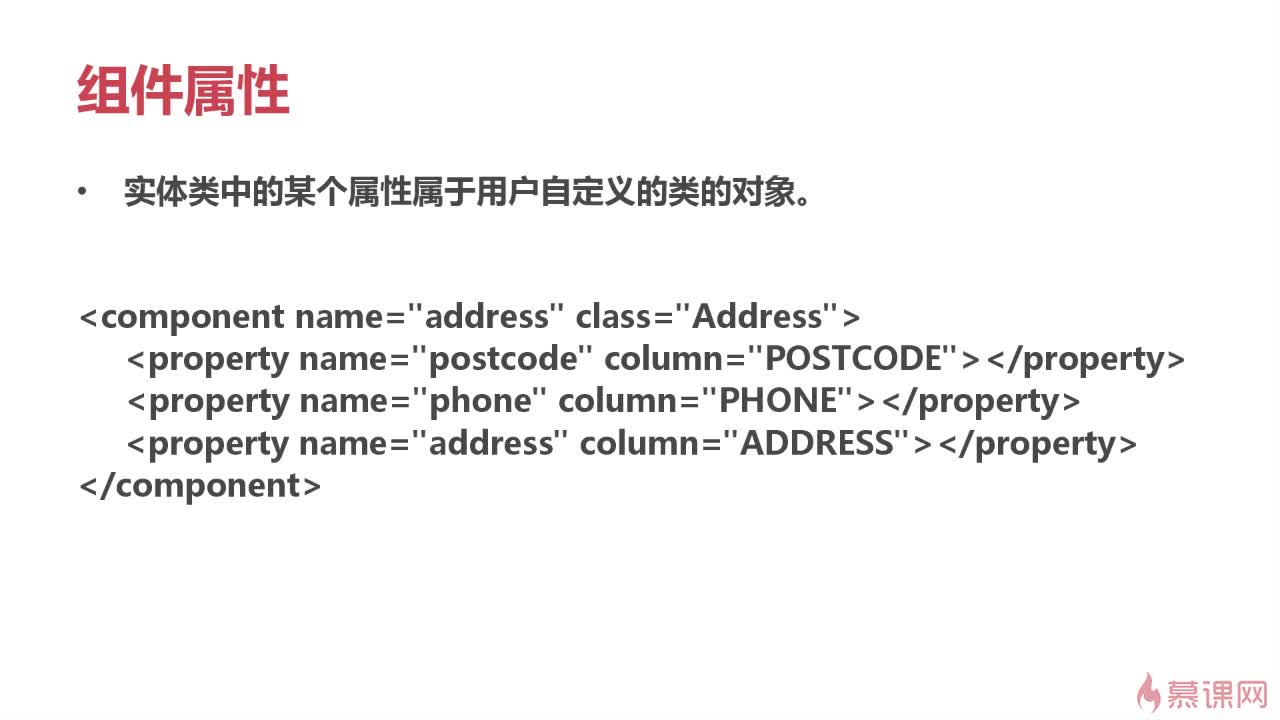
session.get()：单表查询语句

(2) 单表查询get和load的区别

(a) 在不考虑缓存的情况下，get方法会在调用之后立即向数据库发出sql语句，返回持久化对象。

(b) load方法会在调用后返回一个代理对象，该代理对象只保存了实体对象的id，直到使用对象的非主键属性时才会发出sql语句。

(c) 查询数据库中不存在的数据时，get方法返回null，load方法抛出异常org.hibernate.ObjectNotFoundException



三、 关联映射（联表查询）

1. 一对多映射 -- 主表的配置

(1) com.imooc.entity 实体类，映射文件放在和实体类同一层级包中，映射文件配置一对多的关系

(2) <!-- 配置一对多关联关系 --> -- 要在持久化类中添加一个数组的引用

<set name="students" table="student">

<key column="gid"></key>

<one-to-many class="com.imooc.entity.Student"/>

</set>

1. set元素常用属性

name ：映射类属性的名称，不为空。

table ：关联类的的目标数据表。

lazy ：指定关联对象是否使用延迟加载，默认值为proxy。

inverse ：标识双向关联中被动的一方，默认值为false。

2. 多对一映射 -- 外键子表的配置

(1) one to many 和 many to one 需要同时在双方配置

(2) <!-- 配置多对一关联关系 -->

<many-to-one name="grade" class="com.imooc.entity.Grade"

column="gid" cascade="all" />

(3) inverse属性 -- 防止输出重复的SQL语句

<set>节点的inverse属性指定关联关系的控制方向，默认由one方来维护；

关联关系中，inverse="false"则为主动方，由主动方负责维护关联关系；

在一对多关联中，只能设置one方的inverse为true，这将有助于性能的改善。

1. cascade属性

