

第一章 ORM 概述

目标:

- 理解ORM
- 熟悉Hibernate 的作用

软件分层的优点

1. 伸缩性
2. 可维护性
3. 可扩展性
4. 可重用性
5. 可管理性

Hibernate是一个优秀的持久化框架

什么是持久化？

瞬时状态：

保存在内存的程序数据，程序退出后，数据就消失了，称为瞬时状态

内存

姓名：张三
密码：123
性别：男

持久状态：

保存在磁盘上的程序数据，程序退出后依然存在，称为程序数据的持久状态

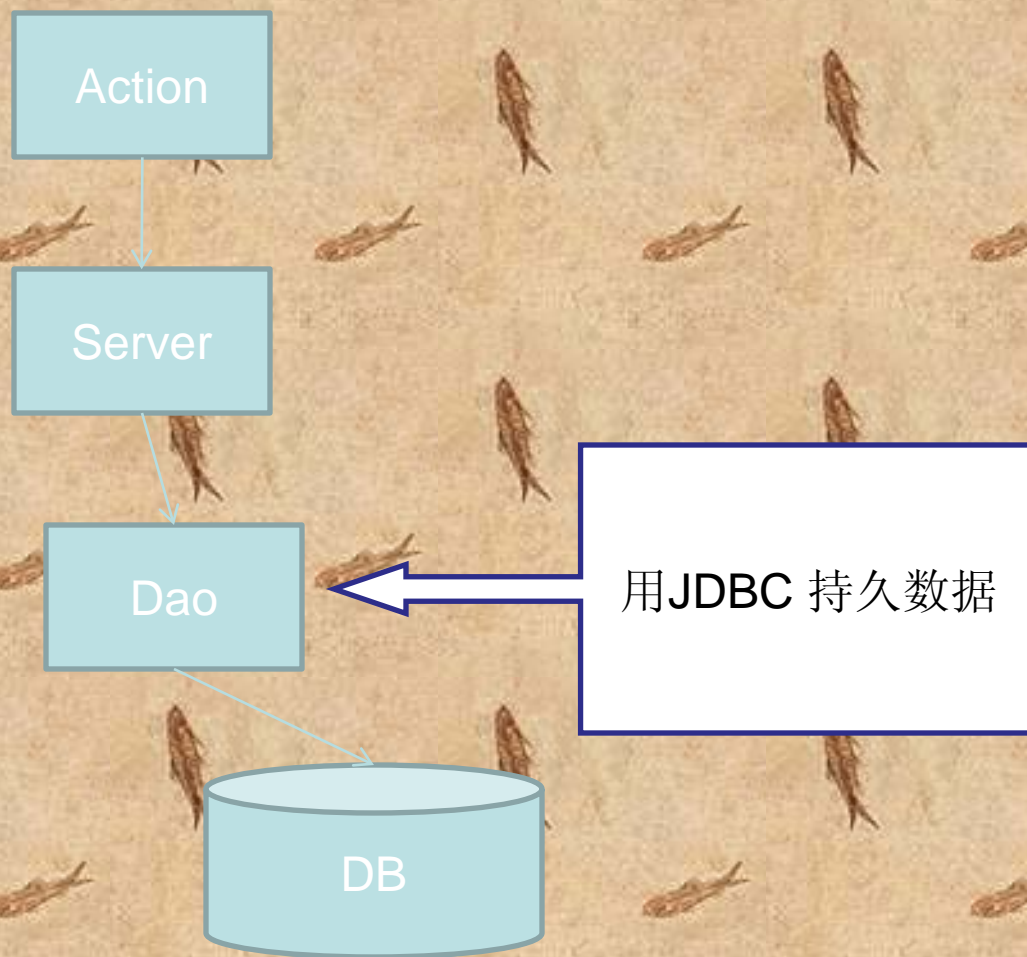
用JDBC完成数据在持久和瞬时状态间的转换：

```
...  
***.execute ("... ")  
...
```

持久化：

将程序数据在瞬时状态和持久状态之间转换的机制

磁盘



Hibernate应用程序的结构

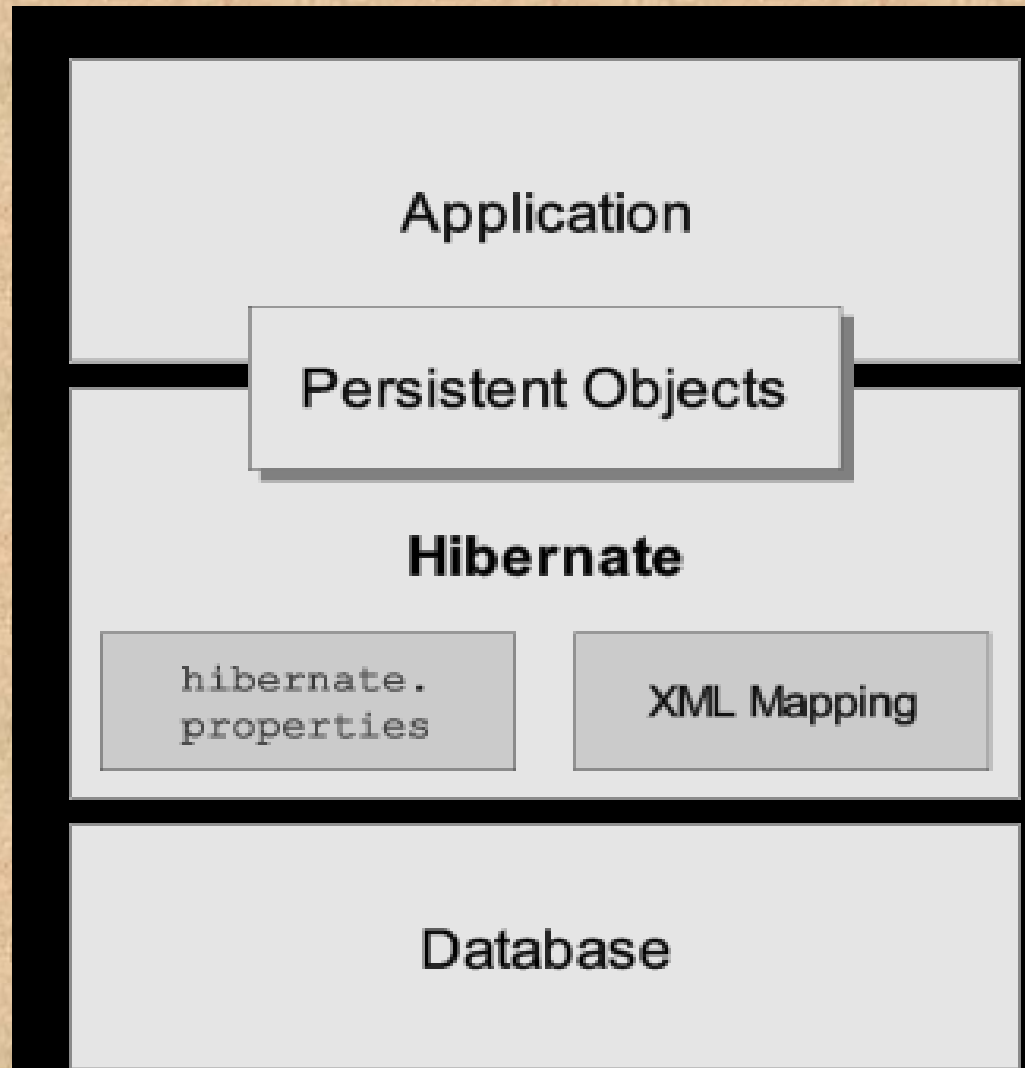
(1) Application: 应用

(2) Persistent Object: 持久化对象

(3) hibernate.properties: Hibernate
属性文件/Hibernate配置文件

(4) XML Mapping: Hibernate映射文
件

(5) Database: 数据库



JDBC与Hibernate的比较

相同点:

- 两者都是JAVA的数据库操作中间件。
- 两者对于数据库进行直接操作的对象都不是线程安全的，都需要及时关闭。
- 两者都可以对数据库的更新操作进行显式的事务处理。

不同点:

- 使用的SQL语言不同: **JDBC**使用的是基于关系型数据库的标准SQL语言, **Hibernate**使用的是HQL(**Hibernate** query language)语言
- 操作的对象不同: **JDBC**操作的是数据, 将数据通过SQL语句直接传送到数据库中执行, **Hibernate**操作的是持久化对象, 由底层持久化对象的数据更新到数据库中。
- 数据状态不同: **JDBC**操作的数据是“瞬时”的, 变量的值无法与数据库中的值保持一致, 而**Hibernate**操作的数据是可持久的, 即持久化对象的数据属性的值是可以跟数据库中的值保持一致的。

• 什么是ORM?

ORM（对象-关系映射）：
完成对象数据到关系型数据映射的机制称为对象-关系映射，简称ORM。

映射信息

类: User

表: TBL_User

属性	字段
name	user_name
sex	user_sex
pw	user_pw

表现层

业务逻辑层

持久化层

TBL_User

对象数据 (O)
User对象

name: 张三
sex: 男
pw: 123

关系型数据
(R)

小结

- 是一个主流的持久化框架
 - 在JDBC基础上进行分装
 - 只需要少量代码就可以完成持久化工作
- 是一个优秀的ORM（对象-关系映射）机制
 - 通过映射文件保存映射信息
 - 在业务层以面向对象的方式编程，不用考虑数据保存形式

3个准备

1) 添加Hibernate库到项目

2-1) : 添加配置文件 -- hibernate.cfg.xml

```
<session-factory>
  <property name="connection.url">
    jdbc:microsoft:sqlserver://localhost:1433;Database=pubs
  </property>
  <property name="connection.username">sa</property>
  <property name="connection.password">pwd</property>
  <property name="connection.driver_class">
    com.microsoft.jdbc.sqlserver.SQLServerDriver
  </property>
  <property name="dialect">
    org.hibernate.dialect.SQLServerDialect
  </property>
  <property name="show_sql">true</property>

  <mapping resource="com/aptech/jb/entity/User.hbm.xml" />
</session-factory>
```

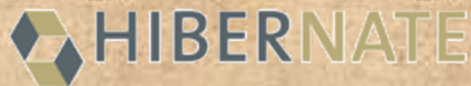
2-2) : 添加配置文件 -- hibernate.cfg.xml

- **<session-factory>**
 - <property name="dialect">**
org.hibernate.dialect.SQLServerDialect
 - </property>**
 - <property**
name="connection.datasource">**java:/comp/env/pubs****</property**
y>
 - <property**
name="session_factory_name">hibernate**</property>**
 - <mapping resource="com/aptech/jb/entity/User.hbm.xml" />**
- **</session-factory>**

准备3:添加实体类和映射文件 (UserInfo.hbm.xml)

使用Hibernate的7个步骤:





根据主键加载

```
Session session = sf.openSession();  
User user = (User)session.get(User.class, id);  
session.close();  
System.out.println(user.getUname() + "," +  
user.getUpass());
```

没有更新数据，
不进行事务控制

修改

```
tx = session.beginTransaction();  
User user = this.testLoad(id);  
user.setUname("new Name");  
session.update(user);  
tx.commit();
```

```
tx = session.beginTransaction();  
User user = this.testLoad(id);  
session.delete(user);  
tx.commit();
```

Hibernate

使用Hibernate实现：

配置Hibernate关联自动加载区县对应的街道

配置Hibernate关联并实现：

- 1、增加一个公司：“CSTP”
- 2、增加CSTP下属的三个部门
- 3、删除“教学”
- 4、划“财务”到红旗渠
- 5、删除红旗渠

— 配置Hibernate多对多关联，实现某OA系统项目和人员对照关系的管理