

# Hibernate

Hibernate概述及入门

讲师：李兴华

更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 本章内容

- Hibernate体系结构
- Hibernate的ORM映射技术



更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



ORMapping：对象关系映射

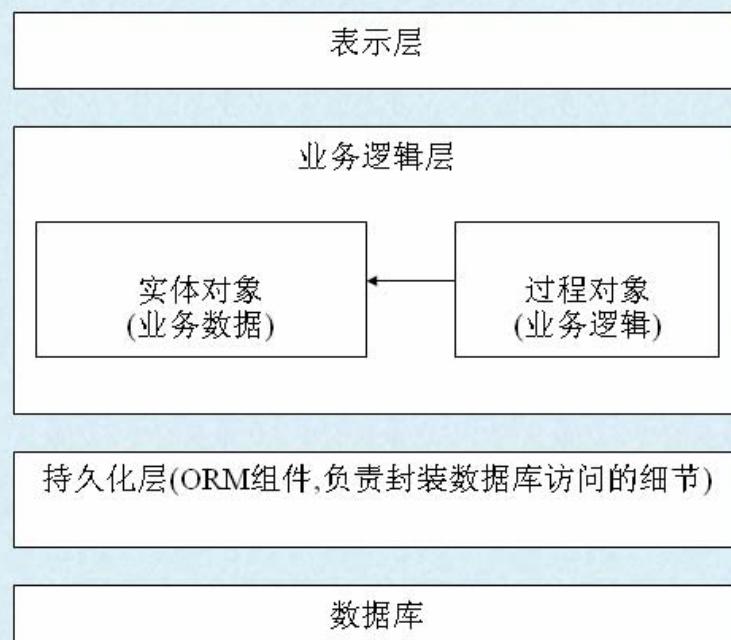
操作数据库使用 SQL 语句

通过一个类的操作 来代表 数据库中的操作

类中的方法 → SQL 语句

Hibernate 是一个 ORMapping 的实现，所以，主要功能就是操作数据库（以对象的形式操作数据库）

## ORMapping模式



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## O/R Mapping的优点

- 提高生产率 (Productivity)
- 可维护性 (Maintainability)
- 更好性能 (Performance)
- 厂商独立性 (Vendor independence)



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 基于RDB的持久层可选方案

	优点	缺点
SQL/JDBC	成熟, 流行, 使用DAO模式	代码烦杂, 可读性差, 维护困难, 移植困难
Entity Bean	CMP (EJB1.1之后), 未来的EJB3	错误的设计。不可移植, 依赖性强, 不可序列化, 不支持多态的关联查询
JDO	简单、透明、标准	不够成熟



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 基于RDB的持久层可选方案

	优点	缺点
Apache OJB	性能、稳定性，属于Apache 基金组织	文档资源太少，支持标准太多成了负担（ODMG 3.0， JDO1.0）
iBATIS	可以控制更多的数据库操作细节。实用于遗留系统的改造和对既有数据库的复用。	持久层封装不够彻底，只是一个DBHelper。
Hibernate	成熟、流行、功能强大。并逐渐发展成Java 持久层事实上的标准。	不够透明



更多学习资源尽在

**www.MLDN.cn**

## Hibernate概述

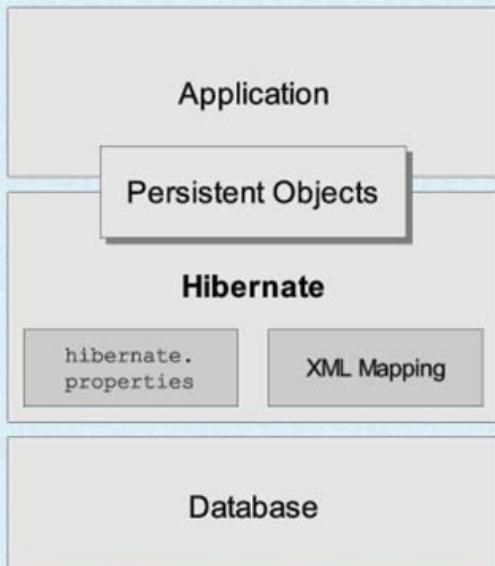
- Hibernate是非常优秀、成熟的O/R Mapping框架。它提供了强大的对象和关系数据库映射以及查询功能。
- Hibernate优势
  - 开源(LGPL): <http://www.hibernate.org/>
  - 成熟
  - 流行(约13 000 downloads/month)
  - 自定义API



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



# Hibernate概览（1）



Hibernate开发步骤：

- 1、持久化类的设计
- 2、持久化类和关系数据库的映射
- 3、应用的开发



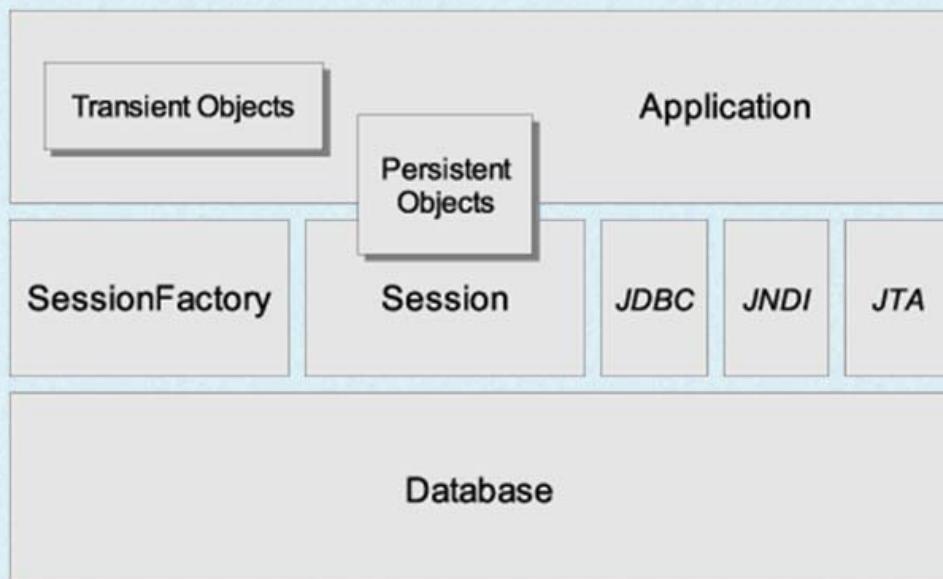
更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



持久化类：ORM的主要特点：  
对象 → 数据库  
类和数据库关系的映射  
应用程序的开发

\

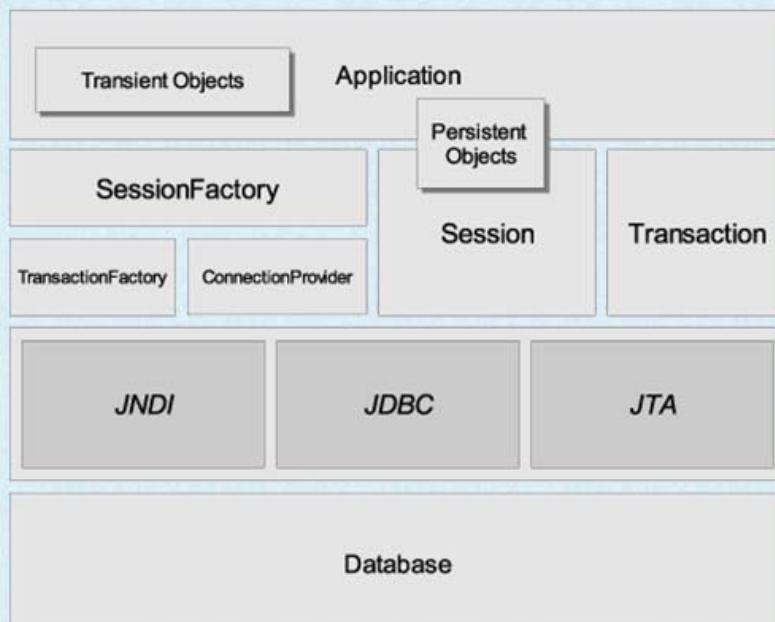
## Hibernate概览（2）



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



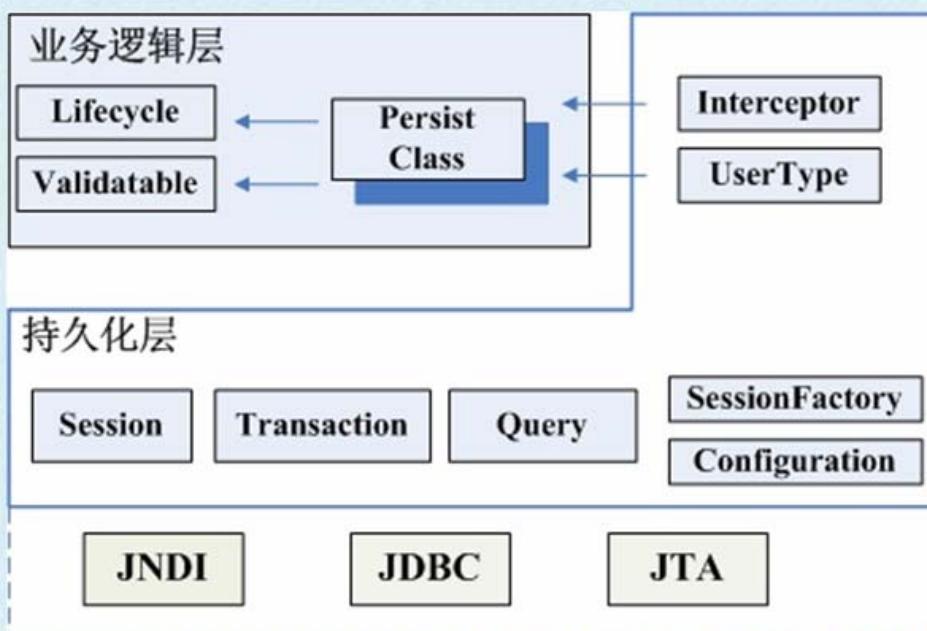
## Hibernate概览（3）



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



# Hibernate核心接口



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



今天的主要关注点：

Session、SessionFactory、Configuration  
Transaction、Query

## Hibernate映射配置

**MiddleGen                    hbm2java**

数据库定义

HBM映射

POJO类

**SchemaExport**

**XDoclet**



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



- POJO 类：只包含 setter 及 getter 方法的类
- HBM 映射：描述类和表之间关系

## POJO

- Hibernate最好的使用方法是使用普通的Java对象(Plain Old Java Objects ,就是POJOs,有时候也称作Plain Ordinary Java Objects) 这种编程模型来进行持久化。一个POJO很像JavaBean,属性通过getter和setter方法访问, 对外隐藏了内部实现的细节。



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



# 第一个Hibernate程序

- 为如下表完成Hibernate映射

```
CREATE TABLE person
(
    id varchar(32) not null primary key,
    name varchar(30) not null ,
    password varchar(30) not null,
    sex varchar(2),
    email varchar(30)
)
```



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



同样使用 Oracle 数据库

数据库创建脚本：

```
DROP TABLE person ;
```

```
CREATE TABLE person
(
    id varchar(32) not null primary key ,
    name varchar(20) not null ,
    password varchar(20) not null ,
    sex varchar(2) ,
    email varchar(30)
);
```

```
-- 事务提交:  
commit ;
```

使用 Eclipse 3.1 + MyEclipse 进行开发

对 MyEclipse 进行配置：

使 Eclipse 中的 MyEclipse 插件能够生成 HBM 文件

The screenshot shows a web page with the MLDN logo and '魔乐在线' text in the top left, and 'IT电子教育门户' in the top right. The main content area has a blue header with 'Person.hbm.xml'. Below the header is the XML code for the mapping file:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 2.0//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd" >
<hibernate-mapping>
<class name="lxh.demo.hibernate.Person" table="person">
  <id name="id" type="java.lang.String" column="id">
    <generator class="uuid.hex" />
  </id>
  <property name="name" type="java.lang.String" column="name" length="30"/>
  <property name="password" type="java.lang.String" column="password" length="30"/>
</class>
</hibernate-mapping>
```

At the bottom of the page, there is a banner with the text '更多学习资源尽在 www.MLDN.cn' and the MLDN logo.

hibernate.cfg.xml：对 Hibernate 环境配置的：包括使用的数据库或数据源

方言：所要使用的数据库类型

POJO 类的名称最好与表名称一致：

表 person，POJO 类名称：Person

建立映射文件

主键生成方式：

assigned：指派，即由用户自行管理

// HBM 根元素，指定 POJO 类所在的包

<hibernate-mapping package="org.lxh.hibernate.demo01">

// 指定Person类与PERSON表的映射

```
<class name="Person" table="PERSON">
    // name表示Person类中的属性名字
    // column: 表示表中的字段名字
    // type: 类型
    <id name="id" column="ID" type="string">
        // 主键的生成方式: assigned: 表示由用户自行管理
        <generator class="assigned"/>
    </id>
    // 表示Person类中与表中其他字段的映射
    // name: Person类中的属性名称
    // column: 表中的字段名称
    // type: 类型
    // not-null: 不允许为空
    <property name="name" column="NAME" type="string" not-null="true" />
    <property name="password" column="PASSWORD" type="string" not-null="true" />
    <property name="sex" column="SEX" type="string" />
    <property name="email" column="EMAIL" type="string" />
</class>

</hibernate-mapping>
```

此文件完成 Person 类到 PERSON 表的关系

取得操作数据库的 Session:

Session 是一个接口，必须通过其他类实例化，格式是固定的

```
// 找到 Hibernate 配置
Configuration config = new Configuration().configure();
// 从配置中取出 SessionFactory
SessionFactory factory = config.buildSessionFactory();
// 从 SessionFactory 中取出一个 Session
this.session = factory.openSession();
```

通过正常配置之后，发现数据无法存入

事务处理，在插入完成之后，必须将事务提交： Transaction

使用 Hibernate 操作数据库，代码量很少，由 Hibernate 完成

用户就感觉像使用对象一样去使用数据库

所有的操作过程都是通过 POJO 类完成

主键生成方式: assigned: 指派

- Sequence: 自动增长的数据段
- uuid.hex: 生成一个 32 位，不会重复的主键

## Configuration

- 概述： Configuration 类负责管理Hibernate的配置信息。它包括如下内容：
  - Hibernate运行的底层信息：数据库的URL、用户名、密码、 JDBC驱动类，数据库 Dialect,数据库连接池等。
  - Hibernate映射文件 (\*.hbm.xml) 。

更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



## SessionFactory

- 概述：应用程序从**SessionFactory**（会话工厂）里获得**Session**(会话)实例。它在多个应用线程间进行共享。通常情况下，整个应用只有唯一的一个会话工厂——例如在应用初始化时被创建。然而，如果你使用**Hibernate**访问多个数据库，你需要对每一个数据库使用一个会话工厂。
- 会话工厂缓存了生成的SQL语句和**Hibernate**在运行时使用的映射元数据。



更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



## Session(会话)

- 概述：

- Session不是线程安全的，它代表与数据库之间的一次操作，它的概念介于Connection和Transaction之间。
- Session也称为持久化管理器，因为它是与持久化有关的操作接口。
- Session通过SessionFactory打开，在所有的工作完成后，需要关闭。
- 它与Web层的HttpSession没有任何关系。



更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



Query 对象、Transaction 对象，都是通过 session 取得的

## Transaction(事务)

- 它将应用代码从底层的事务实现中抽象出来——这可能是一个JDBC事务，一个JTA用户事务或者甚至是一个公共对象请求代理结构(CORBA)——允许应用通过一组一致的API控制事务边界。这有助于保持Hibernate应用在不同类型的执行环境或容器中的可移植性。
- 使用Hibernate进行操作时（增、删、改）必须显示的调用Transaction（默认：autoCommit=false）



更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



## 本章重点

- 掌握OR-Mapping基本原理
- 了解Hibernate体系结构
- 掌握Configuration、SessionFactory、Session、Query、Transaction的使用



更多学习资源尽在

[www.MLDN.cn](http://www.MLDN.cn)



HQL: Hibernate 查询语言: