

一.Hibernate所需的依赖:

主要依赖:hibernate core mysql junit

```
1 <!-- hibernate core mysql junit -->
2 <properties>
3 <hibernate.version>4.3.11.Final</hibernate.version>
4 <mysql.version>5.1.39</mysql.version>
5 <junit.version>4.12</junit.version>
6 <lombok.version>1.18.6</lombok.version>
7
8 </properties>
9
10 <dependencies>
11 <!-- hibernate-core -->
12 <dependency>
13 <groupId>org.hibernate</groupId>
14 <artifactId>hibernate-core</artifactId>
15 <version>${hibernate.version}</version>
16 <!-- 4 重大版本更新 3 小版本的更新 11 4.3 版本进行了多少次的修复 -->
17 </dependency>
18
19 <dependency>
20 <groupId>mysql</groupId>
21 <artifactId>mysql-connector-java</artifactId>
22 <version>${mysql.version}</version>
23 </dependency>
24
25 <dependency>
26 <groupId>org.projectlombok</groupId>
27 <artifactId>lombok</artifactId>
28 <version>${lombok.version}</version>
29 </dependency>
30
31
32 <dependency>
```

```
33 <groupId>junit</groupId>
34 <artifactId>junit</artifactId>
35 <version>${junit.version}</version>
36 <scope>test</scope>
37 </dependency>
38 </dependencies>
```

二.hibernate配置文件在工程的src/main/resource下,新建other,选第一个hibernate

准备Hibernate3-2

示例

■ 编写Hibernate配置文件

- ◆ 用于配置数据库连接
- ◆ 运行时所需的各种特性
- ◆ 一般命名为 “hibernate.cfg.xml”

```
<property name="connection.url">
    jdbc:oracle:thin:@127.0.0.1:1521:orcl</property>
<property name="connection.username">scott</property>
<property name="connection.password">tiger</property>
<property name="connection.driver_class">
    oracle.jdbc.OracleDriver</property>
<property name="dialect">
    org.hibernate.dialect.Oracle10gDialect</property>
<property name="show_sql">true</property>
<property name="format_sql">true</property>
<property name="current_session_context_class">thread</property>
```



演示示例1：搭建Hibernate环境

10/40

拾建1.mp4

Session factory name: sessionFactory

[Get values from Connection](#)

Database dialect: MySQL

Driver class: com.mysql.jdbc.Driver

Connection URL: jdbc:mysql://localhost:3306/shop

Default Schema:

Default Catalog:

Username: root

Password: root

☐ Create a console configuration

1.0X

然后创建一个控制台,勾住,完成后,

```
hibernate01/pom.xml hibernate.cfg.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
5 <hibernate-configuration>
6     <session-factory name="sessionFactory">
7         <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
8         <property name="hibernate.connection.password">root</property>
9         <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/shop</property>
10        <property name="hibernate.connection.username">root</property>
11        <property name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>
12        <property name="show_sql">true</property>
13        <property name="format_sql">true</property>
14        <property name="current_session_context_class">org.hibernate.context.internal.NoContext</property>
15    </session-factory>
16 </hibernate-configuration>
```

?使用current_session_context_class和openSession的区别:

Hibernate openSession() 和 getCurrentSession()的区别

方式说明

- 1.采用getCurrentSession()创建的Session会绑定到当前的线程中去、而采用OpenSession()则不会。
- 2.采用getCurrentSession()创建的Session在commit或rollback后会自动关闭,采用OpenSession()必须手动关闭。
- 3.采用getCurrentSession()需要在Hibernate.cfg.xml配置文件中加入如下配置:

如果是本地事物,及JDBC一个数据库:

```
<property name="Hibernate.current_session_context_class">thread</property>
```

如果是全局事物,及jta事物、多个数据库资源或事物资源:

```
<property name="Hibernate.current_session_context_class">jta</property>
```

总之:

getCurrentSession () 使用当前的session

openSession() 重新建立一个新的session

hibernate交给spring 管理的时,则首选getCurrentSession ()

三.创建持久化类和映射关系

定义持久化类(也称实体类),实现java.io.Serializable接口,添加默认构造方法

因此,要配置映射文件(*.hbm.xml),需要新建new,然后other选hibernate的最后一项,选hbm文件

```
<hibernate-mapping>
  <class name="cn.hibernatedemo.entity.Dept" table="`DEPT`">
    <id name="deptNo" column="`DEPTNO`" type="java.lang.Byte">
      <generator class="assigned" />
    </id>
    <property name="deptName" type="java.lang.String">
      <column name="`DNAME`"></column>
    </property>
    <property name="location" type="java.lang.String" column="`LOC`"/>
  </class>
</hibernate-mapping>
```

?在映射文件中id主键需要改increment

hibernate 常见的主键策略有哪些

1.assigned 主键由外部程序负责生成，在 `save()` 之前必须指定一个。Hibernate不负责维护主键生成。与Hibernate和底层数据库都无关，可以跨数据库。在存储对象前，必须要使用主键的setter方法给主键赋值，至于这个值怎么生成，完全由自己决定，这种方法应该尽量避免

2.increment

由Hibernate从数据库中取出主键的最大值（每个session只取1次），以该值为基础，每次增量为1，在内存中生成主键，不依赖于底层的数据库，因此可以跨数据库。

3.sequence

采用数据库提供的sequence机制生成主键，需要数据库支持sequence。如Oracle、DB2、SAP DB、PostgreSQL、McKoi中的sequence。MySQL这种不支持sequence的数据库

4.identity

identity由底层数据库生成标识符。identity是由数据库自己生成的，但这个主键必须设置为自增长，使用identity的前提条件是底层数据库支持自动增长字段类型，如DB2、SQL Server、MySQL、Sybase和HypersonicSQL等，Oracle这类没有自增字段的则不支持。

5.native

native由hibernate根据使用的数据库自行判断采用identity、hilo、sequence其中一种作为主键生成方式，灵活性很强。如果能支持identity则使用identity，如果支持sequence则使用sequence。**6.uuid**: 适用于char、varchar类型的作为主键。

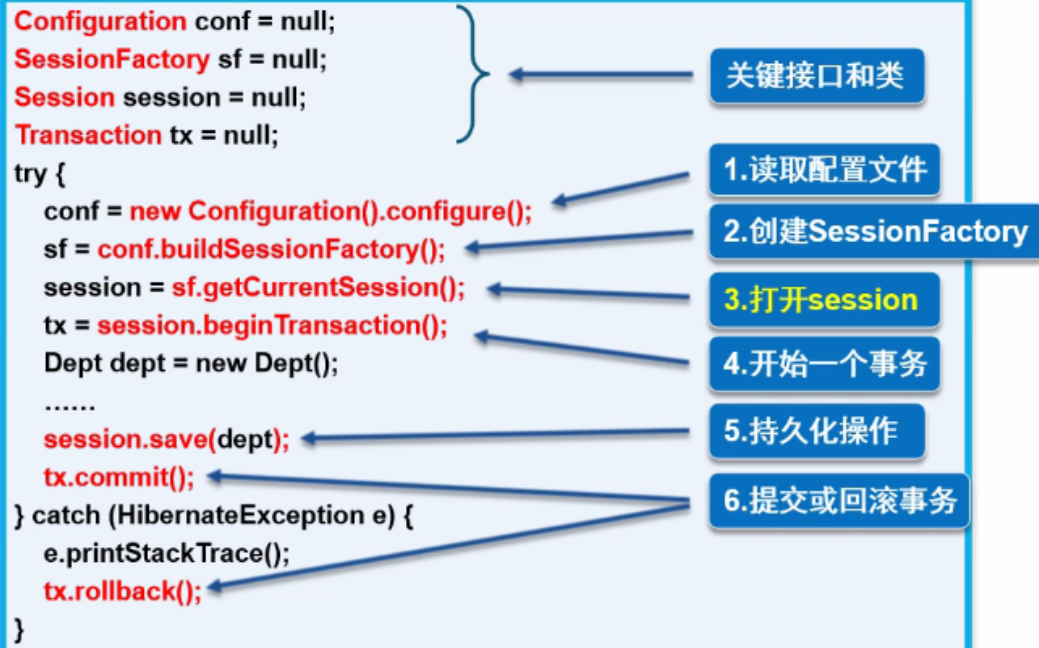
四.引入

◆向hibernate.cfg.xml文件中配置映射文件

```
<session-factory>
    <!-- 省略其他配置 -->
    <mapping resource="cn/hibernatedemo/entity/Dept.hbm.xml" />
</session-factory>
```

五.api

使用Hibernate API 2-2



我们可以新建测试类,对此进行测试.但是其中buildSessionFactory()以过时.

```
1  @Test
2  public void test() {
3      Configuration cfg=new Configuration().configure();
4      SessionFactory sf=cfg.buildSessionFactory();
5      Session session=sf.getCurrentSession();
6
7      //对数据库操作
8      //通过id查信息
9
10     Transaction tx=session.beginTransaction();
11     Role r=(Role) session.get(Role.class,1L);
12     System.out.println(r.toString());
13     tx.commit();
14 }
```