

Maven 3 Exemple Hibernate 3.6 Oracle 11g (mappage XML)

maven hibernate oracle xml mapping

Dans cet article, nous vous montrons comment intégrer **Maven3** , **Hibernate3.6** et **Oracle11g** ensemble. À la fin de cet article, vous allez créer un projet Java avec Maven et insérer un enregistrement dans la base de données Oracle via Hibernate Framework.

Outils

1. Maven 3.0.3
2. JDK 1.6.0__13
3. Hibernate 3.6.3.final
4. Oracle 11g

1. Création de table

Script Oracle SQL pour créer une table « **DBUSER** » dans la base de données.

```
CREATE TABLE DBUSER (  
  USER__ID      NUMBER (5)      NOT NULL,  
  USERNAME      VARCHAR2 (20)   NOT NULL,  
  CREATED__BY   VARCHAR2 (20)   NOT NULL,  
  CREATED__DATE DATE            NOT NULL,  
  PRIMARY KEY ( USER__ID )  
)
```

2. Créer un projet avec Maven

Utilisez Maven pour créer une structure de projet standard.

```
mvn archetype:generate -DgroupId=com.mk Yong -DartifactId=HibernateExample  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

3. Maven vers Eclipse IDE

Convertissez le projet généré par Maven en projet Eclipse et importez-le dans votre IDE Eclipse.

```
mvn eclipse:eclipse
```

4. Ajouter Hibernate et Oracle Dependency

Mettez à jour votre fichier `pom.xml` et ajoutez toutes les dépendances associées.

1. Vous avez besoin d'un «référentiel JBoss» déclaré pour le dernier jarre Hibernate et

sa dépendance.

1. Pour le pilote Oracle JDBC, vous devez l'installer dans votre navigateur local. référentiel manuellement.

- Pour le pilote JDBC Oracle ** Lisez ce guide - [Comment ajouter un pilote JDBC Oracle dans votre repository local Maven](#)

Fichier: pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mkyong.common</groupId>
  <artifactId>HibernateExample</artifactId>
  <packaging>jar</packaging>
  <version>1.0</version>
  <name>HibernateExample</name>
  <url>http://maven.apache.org</url>

  <!-- JBoss repository for Hibernate -->
  <repositories>
    <repository>
      <id>JBoss repository</id>
      <url>http://repository.jboss.org/nexus/content/groups/public/</url>
    </repository>
  </repositories>

  <dependencies>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
```

```

        <version>4.8.2</version>
        <scope>test</scope>
    </dependency>

    <!-- ORACLE JDBC driver, need install yourself -->
    <dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc6</artifactId>
        <version>11.2.0</version>
    </dependency>

    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>3.6.3.Final</version>
    </dependency>

    <dependency>
        <groupId>javassist</groupId>
        <artifactId>javassist</artifactId>
        <version>3.12.1.GA</version>
    </dependency>

</dependencies>
</project>

```

5. Modèle de fichier de mappage Hibernate (hbm)

Créez un fichier de mappage Hibernate XML et une classe Model pour la table “**DBUSER**”.

- Créez le fichier “`DBUser.hbm.xml`” suivant et placez-le sous “**src/main/resources/com/mkyong/user**”.
- Note ** + Créez le dossier s’il n’existe pas.

Fichier: DBUser.hbm.xml

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

```

```

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="com.mkyong.user.DBUser" table="DBUSER">
    <id name="userId" type="int">
      <column name="USER__ID" precision="5" scale="0"/>
      <generator class="assigned"/>
    </id>
    <property name="username" type="string">
      <column name="USERNAME" length="20" not-null="true"/>
    </property>
    <property name="createdBy" type="string">
      <column name="CREATED__BY" length="20" not-null="true"/>
    </property>
    <property name="createdDate" type="date">
      <column name="CREATED__DATE" length="7" not-null="true"/>
    </property>
  </class>
</hibernate-mapping>

```

- Créez un fichier «` DBUser.java` » et placez-le sous « **src/main/java/com/mkyong/user/** »

Fichier: DBUser.java

```

package com.mkyong.user;

import java.util.Date;
/** **
** Dbuser generated by hbm2java
** /public class DBUser implements java.io.Serializable {

    private int userId;
    private String username;
    private String createdBy;
    private Date createdDate;

    public DBUser() {
    }

```

```
public DBUser(int userId, String username, String createdBy,
               Date createdDate) {
    this.userId = userId;
    this.username = username;
    this.createdBy = createdBy;
    this.createdDate = createdDate;
}

public int getUserId() {
    return this.userId;
}

public void setUserId(int userId) {
    this.userId = userId;
}

public String getUsername() {
    return this.username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getCreatedBy() {
    return this.createdBy;
}

public void setCreatedBy(String createdBy) {
    this.createdBy = createdBy;
}

public Date getCreatedDate() {
    return this.createdDate;
}
```

```

    public void setCreatedDate(Date createdDate) {
        this.createdDate = createdDate;
    }
}

```

- Note ** Vous pouvez être intéressé par cet article - [Outils Eclipse Hibernate pour générer automatiquement des fichiers de mapping Hibernate](#) .

6. Fichier de configuration Hibernate

Créez un fichier de configuration Hibernate « **hibernate.cfg.xml** » et placez-le à la racine du dossier des ressources, « **src/main/resources/hibernate.cfg.xml** », puis renseignez les détails de votre base de données Oracle. Et mappez le fichier de mappage Hibernate ci-dessus - « **DBUser.hbm.xml** ».

File: hibernate.cfg.xml

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver__class">oracle.jdbc.driver.OracleDriver</property>
        <property name="hibernate.connection.url">jdbc:oracle:thin:@127.0.0.1:1521:MKYONG</property>
        <property name="hibernate.connection.username">mkyong</property>
        <property name="hibernate.connection.password">password</property>
        <property name="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</property>
        <property name="hibernate.default__schema">MKYONG</property>
        <property name="show__sql">>true</property>
        <mapping resource="com/mkyong/user/DBUser.hbm.xml"></mapping>
    </session-factory>
</hibernate-configuration>

```

7. Utilitaire Hibernate

Créez une classe «` HibernateUtil.java` »classique pour prendre en charge la gestion de session Hibernate. Et mettez sous
" **src/main/java/com/mkyong/util/HibernateUtil.java** "

File: HibernateUtil.java

```
package com.mkyong.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            //Create the SessionFactory from hibernate.cfg.xml
            return new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            //Make sure you log the exception, as it might be swallowed
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    public static void shutdown() {
        //Close caches and connection pools
        getSessionFactory().close();
    }

}
```

8. Réviser la structure finale du projet

Révisé-le et la structure de votre projet devrait ressembler à ceci:

9. Codage Hibernate

Mettez à jour “` App.java `”, pour coder Hibernate afin de sauvegarder un enregistrement d'utilisateur factice dans une table “ **DBUSER** ”.

Fichier: App.java

```
package com.mkyong;

import java.util.Date;
import org.hibernate.Session;
import com.mkyong.util.HibernateUtil;
import com.mkyong.user.DBUser;

public class App {
    public static void main(String[]args) {
        System.out.println("Maven + Hibernate + Oracle");
    }
}
```



```
Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();
DBUser user = new DBUser();

user.setUserId(100);
user.setUsername("superman");
user.setCreatedBy("system");
user.setCreatedDate(new Date());

session.save(user);
session.getTransaction().commit();
}
}
```

10. Le lancer

Exécutez votre " App.java "et voyez le résultat dans la vue de la console Eclipse:

Terminé.

Téléchargez-le - lien://wp-content/uploads/2011/04/Maven3-Hibernate3.6-Oracle11-Example.zip[Maven3-Hibernate3.6-Oracle11-Example.zip](8 Ko)

Référence

1. <http://docs.jboss.org/hibernate/core/3.6/reference/en-US/html/> [Hibernate 3.6 documentation]

lien://tag/hibernate/[hibernate]lien://tag/maven/[maven]lien://tag/oracle/[oracle]lien://tag/xml-mapping/[mapping xml]