Hibernate Research

From my research I have learned that hibernate is a tool used to bring in database code into java so it can be better read and ultized by the java language.

<http://www.mkyong.com/tutorials/hibernate-tutorials/>

<http://www.mkyong.com/hibernate/maven-3-hibernate-3-6-oracle-11g-example-xml-mapping/>

In this article, we show you how to integrate **Maven3**, **Hibernate3.6** and **Oracle11g** together. In the end of this article, you will create a Java project with Maven, and insert a record into Oracle database via Hibernate framework.

Tools & technologies used in this article :

1. Maven 3.0.3
2. JDK 1.6.0\_13
3. Hibernate 3.6.3.final
4. Oracle 11g

1. Table Creation

Oracle SQL script to create a “**DBUSER**” table in database.

**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. Create Project with Maven

Use Maven to create a standard project structure.

mvn archetype:generate -DgroupId=com.mkyong -DartifactId=HibernateExample

-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=**false**

**Note**  
More detail, please refer to this [**How to create a Java project with Maven**](http://www.mkyong.com/maven/how-to-create-a-java-project-with-maven/).

3. Maven to Eclipse IDE

Convert the generated Maven based project to Eclipse project, and import it into your Eclipse IDE.

mvn eclipse:eclipse

4. Add Hibernate and Oracle Dependency

Update your pom.xml file, and add all related dependencies.

1. You need declared “JBoss repository” for the latest Hibernate jar and its dependency.
2. For Oracle JDBC driver, you need to install it into your local maven repository manually.

**For Oracle JDBC Driver**  
Read this guide – [**How to add Oracle JDBC driver in your Maven local repository**](http://www.mkyong.com/maven/how-to-add-oracle-jdbc-driver-in-your-maven-local-repository/)

*File : 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. Hibernate Mapping file (hbm) + Model

Create a Hibernate XML mapping file and Model class for table “**DBUSER**“.

– Create following “DBUser.hbm.xml” file and put it under “**src/main/resources/com/mkyong/user**“.

**Note**  
Create the folder if it does not exists.

*File : 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>**

– Create a “DBUser.java” file and put it under “**src/main/java/com/mkyong/user/**”

*File : 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**  
You may interest read this article – [**Eclipse + Hibernate tools to generate Hibernate mapping files automatically**](http://www.mkyong.com/hibernate/how-to-generate-code-with-hibernate-tools/).

6. Hibernate Configuration File

Create a Hibernate configuration file “**hibernate.cfg.xml**” and put it under the root of resources folder, “**src/main/resources/hibernate.cfg.xml**“, and fill in your Oracle database details. And map to above Hibernate mapping file – “**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. Hibernate Utility

Create a classic “HibernateUtil.java” class to take care of Hibernate session management. And put under “**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*

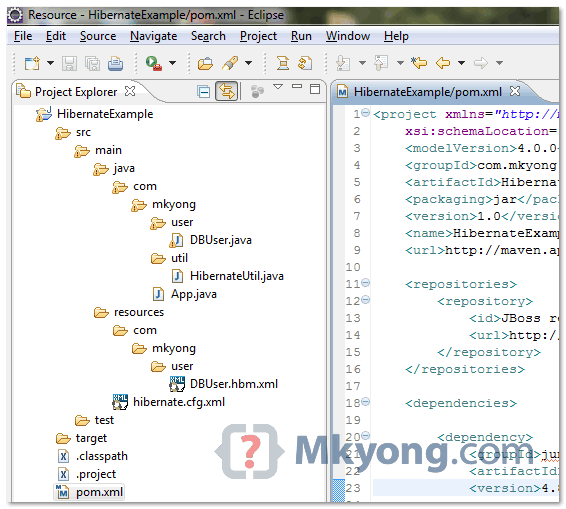
getSessionFactory().close();

}

}

8. Review Final Project Structure

Review it, and your project structure should look like following :



9. Hibernate Coding

Update “App.java“, to code Hibernate to save a dummy user record into a table “**DBUSER**“.

*File : 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);

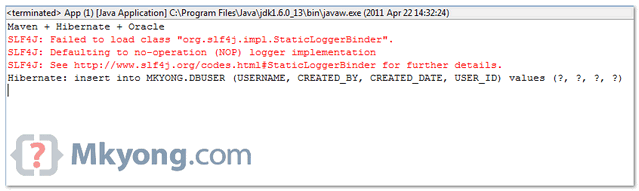
session.getTransaction().commit();

}

}

10. Run It

Run your “App.java“, and see the output in Eclipse console view :



Done.

[**Hibernate**](https://www.hibernate.org/), created by **Gavin King**, known as the best and dominated object/relational persistence (ORM) tool for Java developers (Now is support .NET). It provides many elegant and innovative ways to simplifies the relational database handling task in Java.

**Hibernate** is great at a lot of things, but its something that needs to be used appropriately. In this tutorials, it providing many step by step examples and explanations on using Hibernate3.

P.S Tutorials are updated to Hibernate v3.6.1.Final.

Hibernate Quick Start

Hello World example to experience the Hibernate framework.

* [**Maven 2 + Hibernate 3.2.3 + MySQL 5.0 Example (XML Mapping)**](http://www.mkyong.com/hibernate/quick-start-maven-hibernate-mysql-example/)  
  Hibernate 3.2.3 example in MySQL database, with classic hbm mapping.
* [**Maven 2 + Hibernate 3.2.3 + MySQL 5.0 Example (Annotation)**](http://www.mkyong.com/hibernate/maven-hibernate-annonation-mysql-example/)  
  Hibernate 3.2.3 example in MySQL database, with Hibernate / JPA annotation.
* [**Maven 3 + Hibernate 3.6.3 + Oracle 11g Example (XML Mapping)**](http://www.mkyong.com/hibernate/maven-3-hibernate-3-6-oracle-11g-example-xml-mapping/)  
  Hibernate 3.6 example in Oracle database, with classic hbm mapping.
* [**Maven 3 + Hibernate 3.6.3 + Oracle 11g Example (Annotation)**](http://www.mkyong.com/hibernate/maven-3-hibernate-3-6-oracle-11g-example-annotation/)  
  Hibernate 3.6 example in Oracle database, with Hibernate / JPA annotation.

Hibernate Association (Table Relationship)

How to define table relationship one-to-one, one-to-many, many-to-may in Hibernate.

* [**One-to-One example (XML Mapping)**](http://www.mkyong.com/hibernate/hibernate-one-to-one-relationship-example/)  
  Hibernate one to one example with hbm mapping file.
* [**One-to-One example (Annotation)**](http://www.mkyong.com/hibernate/hibernate-one-to-one-relationship-example-annotation/)  
  Hibernate one to one example with annotation code.
* [**One-to-Many example (XML Mapping)**](http://www.mkyong.com/hibernate/hibernate-one-to-many-relationship-example/)  
  Hibernate one to many example with hbm mapping file.
* [**One-to-Many example (Annotation)**](http://www.mkyong.com/hibernate/hibernate-one-to-many-relationship-example-annotation/)  
  Hibernate one to many example with annotation code.
* [**Many-to-Many example (XML Mapping)**](http://www.mkyong.com/hibernate/hibernate-many-to-many-relationship-example/)  
  Hibernate many to many example (NO extra column in join table) with hbm mapping file.
* [**Many-to-Many example (Annotation)**](http://www.mkyong.com/hibernate/hibernate-many-to-many-relationship-example-annotation/)  
  Hibernate many to many example (NO extra column in join table) with annotation code.
* [**Many-to-Many example – join table + extra column (Annotation)**](http://www.mkyong.com/hibernate/hibernate-many-to-many-example-join-table-extra-column-annotation/)  
  Hibernate many to many example (WITH extra column in join table) with annotation code.
* [**Off Topic : Understand inverse keywork, example and explanation**](http://www.mkyong.com/hibernate/inverse-true-example-and-explanation/)  
  The “inverse” is the most confusing keyword in Hibernate, but you have to understand this clearly in order to fine tune your relationship performance.

Hibernate / JBoss Tools + Eclipse IDE

Learn how to use the Hibernate tools is a must!

* [**Install Hibernate / JBoss Tools in Eclipse IDE**](http://www.mkyong.com/hibernate/how-to-install-hibernate-tools-in-eclipse-ide/)  
  Install Hibernate in your Eclipse IDE.
* [**Generate Hibernate mapping files & annotation with Hibernate Tools**](http://www.mkyong.com/hibernate/how-to-generate-code-with-hibernate-tools/)  
  Auto generate Hibernate codes for you.

Hibernate Logging

How to do logging in Hibernate

* [**Configure logging in Hibernate – SLF4j + Log4j**](http://www.mkyong.com/hibernate/how-to-configure-log4j-in-hibernate-project/)  
  Integrate SLF4j + Log4j with Hibernate.
* [**Configure logging in Hibernate – Logback**](http://www.mkyong.com/hibernate/how-to-configure-logging-in-hibernate-logback/)  
  Integrate Logback with Hibernate.

Hibernate Connection Pool

How to configure database connection pool in Hibernate

* [**Configure C3P0 connection pool in Hibernate**](http://www.mkyong.com/hibernate/how-to-configure-the-c3p0-connection-pool-in-hibernate/)  
  Integrate C3P0 with Hibernate.
* [**Configure DBCP connection pool in Hibernate**](http://www.mkyong.com/hibernate/how-to-configure-dbcp-connection-pool-in-hibernate/)  
  Integrate Apache DBCP with Hibernate.

Hibernate Cascade

Hibernate cascade is use to manage the state of the other side automatically.

* [**Cascade example (save, update, delete and delete-orphan)**](http://www.mkyong.com/hibernate/hibernate-cascade-example-save-update-delete-and-delete-orphan/)  
  Cascade examples in save, update, delete and delete orphan. And the different between delete and delete orphan.
* [**Different between cascade and inverse**](http://www.mkyong.com/hibernate/different-between-cascade-and-inverse/)  
  Many Hibernate developers are confuse about the different between the cascade and inverse, here’s the explanation.
* [**Cascade – JPA & Hibernate annotation common mistake**](http://www.mkyong.com/hibernate/cascade-jpa-hibernate-annotation-common-mistake/)  
  A super easy common annotation mistake made by beginner or experienced Hibernate developers – JPA cascade annotation in Hibernate?

Hibernate Query Language (HQL)

Hibernate own language for the data manipulation, it’s quite similar to database SQL language.

* [**Hibernate Query examples (HQL)**](http://www.mkyong.com/hibernate/hibernate-query-examples-hql/)  
  HQL CRUD examples , select, update , delete and batch insert (no single insert support).
* [**Hibernate parameter binding examples**](http://www.mkyong.com/hibernate/hibernate-parameter-binding-examples/)  
  Bind the parameter into HQL with “Named parameters” and “Positional parameters” methods.
* [**How to embed Oracle hints in Hibernate query**](http://www.mkyong.com/hibernate/how-to-embed-oracle-hints-in-hibernate-query/)  
  A trick to embed the Oracle hints into Hibernate Query to increase the Oracle query performance.

Hibernate Criteria

Hibernate Criteria API is an alternative to Hibernate Query Language (HQL). It’s always a good solution in many optional search criteria.

* [**Hibernate Criteria examples**](http://www.mkyong.com/hibernate/hibernate-criteria-examples/)  
  Criteria examples – basic query, ordering query, restrictions query and paging the result.

Hibernate Native SQL

In some scenarios, Hibernate HQL or Criteria is just not enough to do what you want, here you can use the native database SQL language directly.

* [**Hibernate native SQL queries examples**](http://www.mkyong.com/hibernate/hibernate-native-sql-queries-examples/)  
  A guide to show how to use native SQL in Hibernate.

Hibernate Named Query

Named Query lets developers put the HQL into XML mapping file or annotation for maintainability purpose, you just do not want all your HQL syntax scatter all over the Java code. :)

* [**Hibernate named query examples**](http://www.mkyong.com/hibernate/hibernate-named-query-examples/)  
  Working with named query in XML file and Annotation.

Hibernate Transaction

All things related to Hibernate Transaction

* [**Hibernate Transaction handle example**](http://www.mkyong.com/hibernate/hibernate-transaction-handle-example/)  
  A simple standard example to use Hibernate transaction.

Hibernate Advance Technique

Some Hibernate Advance technique, seldom use but practical skills (data filter and interceptor).

* [**Hibernate data filter example – XML and annotation**](http://www.mkyong.com/hibernate/hibernate-data-filter-example-xml-and-annotation/)  
  Hibernate data filter is used to filter the retrieve data from database, Here’s a guide to work with data filter in XML or Annotation.
* [**Hibernate interceptor example – audit log**](http://www.mkyong.com/hibernate/hibernate-interceptor-example-audit-log/)  
  Hibernate interceptor is used to intercept the Hibernate events like CRUD operations, a detail example of audit log implementation with Hibernate interceptor.

Hibernate Performance

Some tweaks will make your Hibernate run faster :)

* [**dynamic-insert attribute example**](http://www.mkyong.com/hibernate/hibernate-dynamic-insert-attribute-example/)  
  Using dynamic-insert to avoid the include unmodified properties in the SQL INSERT statement.
* [**dynamic-update attribute example**](http://www.mkyong.com/hibernate/hibernate-dynamic-update-attribute-example/)  
  Using dynamic-insert to avoid the include unmodified properties in the SQL UPDATE statement.
* [**Hibernate mutable example (class and collection)**](http://www.mkyong.com/hibernate/hibernate-mutable-example-class-and-collection/)  
  Using mutable keyword to avoid the generate unnecessary SQL statements.
* [**Hibernate – fetching strategies examples**](http://www.mkyong.com/hibernate/hibernate-fetching-strategies-examples/)  
  Hibernate fetching strategies are used to optimize the Hibernate generated select statement, a must learn skill for any Hibernate developers.
* [**Different between session.get() and session.load()**](http://www.mkyong.com/hibernate/different-between-session-get-and-session-load/)  
  Understanding when should use get or load to retrieve the object in order to avoid unnecessary hit to the database.

Integrate Hibernate with Other Frameworks

Examples to integrate Hibernate with other frameworks.

* [**Struts + Hibernate Integration**](http://www.mkyong.com/struts/struts-hibernate-integration-example/)  
  Example to integrate Hibernate with Struts framework.
* [**Struts + Spring + Hibernate Integration**](http://www.mkyong.com/struts/struts-spring-hibernate-integration-example/)  
  Example to integrate Hibernate with Struts and Spring framework together.
* [**Spring + Hibernate Integration**](http://www.mkyong.com/spring/maven-spring-hibernate-mysql-example/)  
  Example to integrate Hibernate with Spring framework.
* [**Spring + Hibernate Integration (Annotation)**](http://www.mkyong.com/spring/maven-spring-hibernate-annotation-mysql-example/)  
  Example to integrate Hibernate with Spring framework (Annotation version).

Hibernate FAQ

Some frequence answer questions :

* [**How to load hibernate.cfg.xml from different directory**](http://www.mkyong.com/hibernate/how-to-load-hibernate-cfg-xml-from-different-directory/)  
  By default, Hibernate looking the hibernate.cfg.xml at the project class path, here’s a guide to load it from specified folder.
* [**How to add Hibernate XML mapping file (hbm.xml) programmatically**](http://www.mkyong.com/hibernate/how-to-add-hibernate-xml-mapping-file-hbm-xml-programmatically/)  
  A tip to load hibernate.cfg.xml programmatically
* [**List of Hibernate database Dialect**](http://www.mkyong.com/hibernate/hibernate-dialect-collection/)  
  List of the dialect collection for different type of database vendors.
* [**show\_sql , format\_sql and use\_sql\_comments**](http://www.mkyong.com/hibernate/hibernate-display-generated-sql-to-console-show_sql-format_sql-and-use_sql_comments/)  
  Configure Hibernate to display the generated SQL statement to console.
* [**How to display hibernate sql parameter values – P6Spy**](http://www.mkyong.com/hibernate/how-to-display-hibernate-sql-parameter-values-solution/)  
  Using P6Sqpy third party library to display the Hibernate SQL parameter value.
* [**How to display hibernate sql parameter values – Log4j**](http://www.mkyong.com/hibernate/how-to-display-hibernate-sql-parameter-values-log4j/)  
  Using Log4j to display the Hibernate SQL parameter value.
* [**How to call store procedure in Hibernate**](http://www.mkyong.com/hibernate/how-to-call-store-procedure-in-hibernate/)  
  It’s not recommend to put business logic into store procedure, never mind, you still allow to call store procedure in Hibernate.
* [**How to use database reserved keyword in Hibernate**](http://www.mkyong.com/hibernate/how-to-use-database-reserved-keyword-in-hibernate/)  
  In some special case, you may need to use the database keyword in your Hibernate class (not recommend), here’s a trick to achieve it.
* [**How to save an image into database**](http://www.mkyong.com/hibernate/hibernate-save-image-into-database/)  
  A tutorial to show how to use Hibernate to save an image into database.

Hibernate Common Errors

Here are list of the common errors messages in Hibernate development.

* [**Unable to insert if column named is keyword, such as DESC**](http://www.mkyong.com/hibernate/hibernate-unable-to-insert-if-column-named-is-keyword-such-as-desc/)
* [**Hibernate – Could not find C3P0ConnectionProvider**](http://www.mkyong.com/hibernate/hibernate-could-not-find-c3p0connectionprovider/)
* [**Hibernate – The type AnnotationConfiguration is deprecated**](http://www.mkyong.com/hibernate/hibernate-the-type-annotationconfiguration-is-deprecated/)
* [**java.lang.ClassNotFoundException : javassist.util.proxy.MethodFilter**](http://www.mkyong.com/hibernate/java-lang-classnotfoundexception-javassist-util-proxy-methodfilter/)
* [**Remember that ordinal parameters are 1-based! – Hibernate Template**](http://www.mkyong.com/hibernate/remember-that-ordinal-parameters-are-1-based-hibernatetemplate/)
* [**org.hibernate.AnnotationException: Unknown Id.generator**](http://www.mkyong.com/hibernate/org-hibernate-annotationexception-unknown-id-generator/)
* [**An AnnotationConfiguration instance is required to use**](http://www.mkyong.com/hibernate/hibernate-error-an-annotationconfiguration-instance-is-required-to-use/)
* [**java.lang.NoClassDefFoundError: org/dom4j/DocumentException**](http://www.mkyong.com/hibernate/hibernate-error-initial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-orgdom4jdocumentexception/)
* [**java.lang.NoClassDefFoundError: org/apache/commons/logging/LogFactory**](http://www.mkyong.com/hibernate/hibernate-error-initial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-orgapachecommonslogginglogfactory/)
* [**java.lang.NoClassDefFoundError: org/apache/commons/collections/SequencedHashMap**](http://www.mkyong.com/hibernate/hibernate-error-initial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-orgapachecommonscollectionssequencedhashmap/)
* [**java.lang.NoClassDefFoundError: net/sf/cglib/proxy/CallbackFilter**](http://www.mkyong.com/hibernate/hibernate-error-initial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-netsfcglibproxycallbackfilter/)
* [**java.lang.NoClassDefFoundError: com/mchange/v2/c3p0/DataSources**](http://www.mkyong.com/hibernate/hibernate-errorinitial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-commchangev2c3p0datasources/)
* [**java.lang.NoClassDefFoundError: org/hibernate/annotations/common/reflection/ReflectionManager**](http://www.mkyong.com/hibernate/hibernate-error-initial-sessionfactory-creation-failed-java-lang-noclassdeffounderror-orghibernateannotationscommonreflectionreflectionmanager/)
* [**java.lang.NoClassDefFoundError: antlr/ANTLRException**](http://www.mkyong.com/hibernate/hibernate-error-exception-in-thread-main-java-lang-noclassdeffounderror-antlrantlrexception/)
* [**java.lang.NoClassDefFoundError: javax/transaction/Synchronization**](http://www.mkyong.com/hibernate/hibernate-error-java-lang-noclassdeffounderror-javaxtransactionsynchronization/)
* [**java.lang.ClassFormatError : Absent Code attribute in method that is not native or abstract in class file …**](http://www.mkyong.com/hibernate/java-lang-classformaterror-absent-code-attribute-in-method-that-is-not-native-or-abstract-in-class-file/)
* [**java.lang.NoSuchMethodError: org.objectweb.asm.ClassWriter**](http://www.mkyong.com/hibernate/java-lang-nosuchmethoderror-org-objectweb-asm-classwriter/)
* [**java.lang.ClassNotFoundException: javax.persistence.Entity**](http://www.mkyong.com/hibernate/java-lang-classnotfoundexception-javax-persistence-entity/)
* [**java.lang.ClassNotFoundException: javax.transaction.TransactionManager**](http://www.mkyong.com/hibernate/java-lang-classnotfoundexception-javax-transaction-transactionmanager/)
* [**java.lang.ClassFormatError : Absent Code attribute in method that is not native or abstract in class file …**](http://www.mkyong.com/hibernate/java-lang-classformaterror-absent-code-attribute-in-method-that-is-not-native-or-abstract-in-class-file/)

Off Topic

* [**Why i choose Hibernate framework for my project?**](http://www.mkyong.com/hibernate/why-i-choose-hibernate-for-my-project/)  
  The reasons why i like to implement Hibernate in the future project.

Hibernate References

* [**Hibernate Official Documentation**](http://docs.jboss.org/hibernate/stable/core/reference/en/html/)
* [**Hibernate Wiki**](http://en.wikipedia.org/wiki/Hibernate_%28Java%29)