

Hibernate

Hibernate ORM (*Object Relational Mapping*) permite a los desarrolladores escribir de una manera más sencilla aplicaciones cuyos datos sobreviven al proceso de la aplicación. *Hibernate* es un *framework* de mapeo Objeto/Relacional (ORM), el cual se encarga de mantener los datos persistentes que son aplicados a base de datos relacionales para la plataforma *Java* (vía *JDBC*). Este permite facilitar el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (*XML*) o anotaciones de las entidades que permiten establecer estas relaciones.

Pasos para la instalación y configuración de Hibernate:

1. Descargar la librería de *Hibernate ORM* a través del siguiente enlace <http://hibernate.org/orm/downloads/> en la última versión disponible. En el siguiente enlace <http://hibernate.org/orm/documentation/> usted podrá además consultar toda la documentación necesaria que necesite sobre *Hibernate*.

2. Una vez que haya descargado la librería, debe descomprimirla y puede almacenarla en el directorio de su preferencia.

3. Es necesario tener instalado un *IDE* (*Integrated Development Enviroment*) para poder trabajar con *Hibernate*, como *Netbeans* o *Eclipse*, en este caso se recomienda utilizar *Eclipse* para desarrollo en *Java*. Posteriormente se cargará *Hibernate* en el *Java Build Path*. Se explicará la instalación y configuración de *Hibernate* utilizando *Eclipse*.

4. En el directorio */lib/* de la librería que usted descargó, va a encontrar todos los **.jar** pertenecientes a *Hibernate*. En el directorio */lib/required/* se encuentran todos los **.jar** necesarios para obtener una aplicación básica en *Java*. El *hibernate-core* es el **.jar** principal de *hibernate*, todos los demás **.jar** son dependencias de *hibernate* pero que también deben ser incluidas para crear un proyecto en *Hibernate*. Estos **.jar** deben incluirse en el nuevo *Java Project* a crear en *Eclipse* a través del *Java Build Path*. Los directorios restantes contenidos dentro del directorio */lib/* contienen otras características de *Hibernate*.

5. Crear un nuevo *Java Project* con un nombre para su proyecto. En la pestaña *Libraries* agregar una librería en la opción *Add Library*, seleccionar la opción *User Library* y en la opción *User Library* agregar una nueva librería con un nombre dado. A esa nueva librería se le deben agregar todos los **.jar** del directorio */lib/required/* mencionado anteriormente con la opción *Add External JARs* en la ubicación de la librería descargada. Luego se debe seleccionar la librería agregada en la lista de librerías de su proyecto y puede finalizar la creación de su proyecto. Si crea el *Java Project* sin especificar la librería a utilizar inicialmente, igual puede modificar esto a través de *Properties* en el *Java Build Path*.

6. Además usted deberá instalar un *driver* para conectarse a la base de datos desde *Java*. Como *DBMS* podría utilizar *MySQL*, *Oracle*, o cualquier otro. En este caso se recomienda utilizar *MySQL*. Ya que *Hibernate* trabaja con *JDBC*, se recomienda descargar el *JDBC Driver for MySQL* (*Connector/J*) del siguiente enlace <http://www.mysql.com/products/connector/> y almacenarlo en el directorio de su preferencia.

7. Seleccionar el *Build Path* de su proyecto creado, para agregar el **.jar** descargado en el

paso anterior para poderse conectar a la base de datos a través de *MySQL*. De igual manera debe seleccionar la opción *Add External JARs* para agregar el **.jar** de *MySQL* a este proyecto. Es recomendable probar la conexión a la base de datos en *MySQL* antes de empezar a trabajar en *Hibernate*.

8. Para trabajar en *Hibernate*, es importante crear el archivo de configuración llamado **hibernate.cfg.xml** en donde usted podrá registrar todos los requerimientos de configuración necesarios, además del mapeo de cada una de las clases que formen parte de su proyecto. Puede guiarse del ejemplo de configuración de *Hibernate* en la sección: **1.1.4. Hibernate Configuration** en la documentación de *Hibernate* mencionada previamente. A continuación se presenta un ejemplo del archivo de configuración **hibernate.cfg.xml** que usted debe crear para que su proyecto en *Hibernate* funcione correctamente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>

        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/database_name</property>
        <property name="connection.username">user_name</property>
        <property name="connection.password">user_password</property>

        <!-- JDBC connection pool (use the built-in) -->
        <property name="connection.pool_size">1</property>

        <!-- SQL dialect -->
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>

        <!-- Enable Hibernate's automatic session context management -->
        <property name="current_session_context_class">thread</property>

        <!-- Disable the second-level cache -->
        <property name="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="show_sql">>true</property>

        <!-- Drop and re-create the database schema on startup -->
        <property name="hbm2ddl.auto">create</property>

        <!-- Mention here all the model classes along with their package name -->
        <!-- Using Annotations -->
        <mapping class="Class_Name"/>
        <!-- Using hbm.xml files -->
        <mapping resource="resources/org/hibernate/tutorial/domain/Class_Name.hbm.xml"/>

    </session-factory>
</hibernate-configuration>
```

El *Session Factory* se encarga de una base de datos en particular. Si usted tiene varias bases de datos, entonces puede utilizar `</session-factory>` en varios archivos de configuración. Inicialmente se debe proveer la configuración necesaria para poder realizar la conexión a la base de datos. Se debe especificar el *driver* de conexión a través de *JDBC* y *MySQL*, ya que estamos trabajando con *Java* y *MySQL*. Luego debe especificarse en el *url* de conexión, el puerto y el nombre de la base de datos creada. Además debe proveerse el nombre de usuario y *password* para realizar la conexión a la base de datos a utilizar.

Luego, se debe especificar el tamaño del *pool* de conexiones `connection.pool_size`, por defecto se utilizará el *built-in*. Además, se debe especificar a través del *dialect* cuál será la base de datos a utilizar, en este caso se indica que se trabajará con *MySQL*. Es importante especificar esta propiedad ya que cada manejador de base de datos trabaja de forma diferente.

Activar la administración sesiones automáticas en *Hibernate* a través de la opción `current_session_context_class`, es bastante útil para la persistencia de contextos. A través de la opción `show_sql` se puede indicar si se desea ver la salida de los comandos en *SQL*. Además a través de la opción `hbm2ddl.auto` se activa la generación automática de esquemas de la base de datos directamente en la base de datos. Este puede tomar valores como: *create*, *update*, *create-drop* o *validate*.

Finalmente lo más importante es agregar los archivos de *mapping* de las clases para que las clases sean persistentes. Existen dos formas:

- En el caso de utilizar anotaciones se agrega `<mapping class="Person"/>` para registrar a la clase como tal.
- En el caso de utilizar archivos *XML* como *hbm.xml* se agrega `<mapping resource="resources/org/hibernate/tutorial/domain/Class_Name.hbm.xml"/>` para registrar el archivo *XML* en donde se encuentra la configuración de la clase en particular.

Es opcional agregar el *Hibernate DTD*, se puede utilizar para auto completar el *mapping* de elementos en *XML* y atributos en el editor o *IDE*.

Adicionalmente, aparte de la documentación que ofrece *Hibernate* en su página principal <http://hibernate.org/orm/>, existen otros tutoriales o ejemplos en línea que pueden servirle de ayuda en el momento de crear un nuevo proyecto.

Se recomienda ver el tutorial de *gontuseries* en *youtube* el cual contiene una lista de vídeos sobre *Hibernate*: <https://www.youtube.com/playlist?list=PLBgMUB7xGcO0cujAlaeDDEOdZkeNZUScM>.