

HIBERNATE

Come ottenere Hibernate

Hibernate è un software molto complesso suddiviso in una serie di moduli/artefatti (archivi jar) in modo da isolare le dipendenze (modularità del software).

Questi moduli/artefatti che compongono l'intero ecosistema Hibernate sono:

- **Hibernate-core**

Hibernate-core è il modulo principale di hibernate (il cuore appunto).

Definisce le sue caratteristiche di ORM e API.

Questo è l'unico modulo davvero necessario per poter utilizzare hibernate.

La maggior parte (ma non tutti) degli altri moduli elencati sotto hibernate-core sono dei moduli di integrazioni con altri sistemi software.

- **Hibernate-envers**

- **Hibernate-spaziale**

- **Hibernate-OSGi**

- **Hibernate-Agrol**

- **Hibernate-c3p0**

- **Hibernate-hikaricp**

- **Hibernate-vibur**

- **Hibernate-Proxool**

- **Hibernate-jcache**
- **Hibernate-EHCache**

Il team di sviluppatori di Hibernate fornisce il rilascio del software di Hibernate

- Nel formato zip
- Nel formato TGZ (Tar GnuZip).

Il formato TGZ è una forma abbreviata della doppia estensione "TAR.GZ". Viene assegnata ai file TAR che sono stati compressi utilizzando l'algoritmo GnuZip (GZIP).

Il formato di archiviazione TAR viene usato per aggregare un insieme di file in un unico pacchetto (archivio) facile da gestire, ma non contiene nessuna compressione di per sé.

Per ridurre le dimensioni dell'archivio, i file.tar sono compressi con l'algoritmo Zip Gnu, risultando un file.tgz finale.

File TGZ sono progettati per l'utilizzo su Unix, Linux e sistemi OS X, ma non sono supportati in ambiente Windows senza alcuni software particolari di terze parti.

Ogni file di rilascio sia che esso sia nel formato zip o tar.gz contiene :

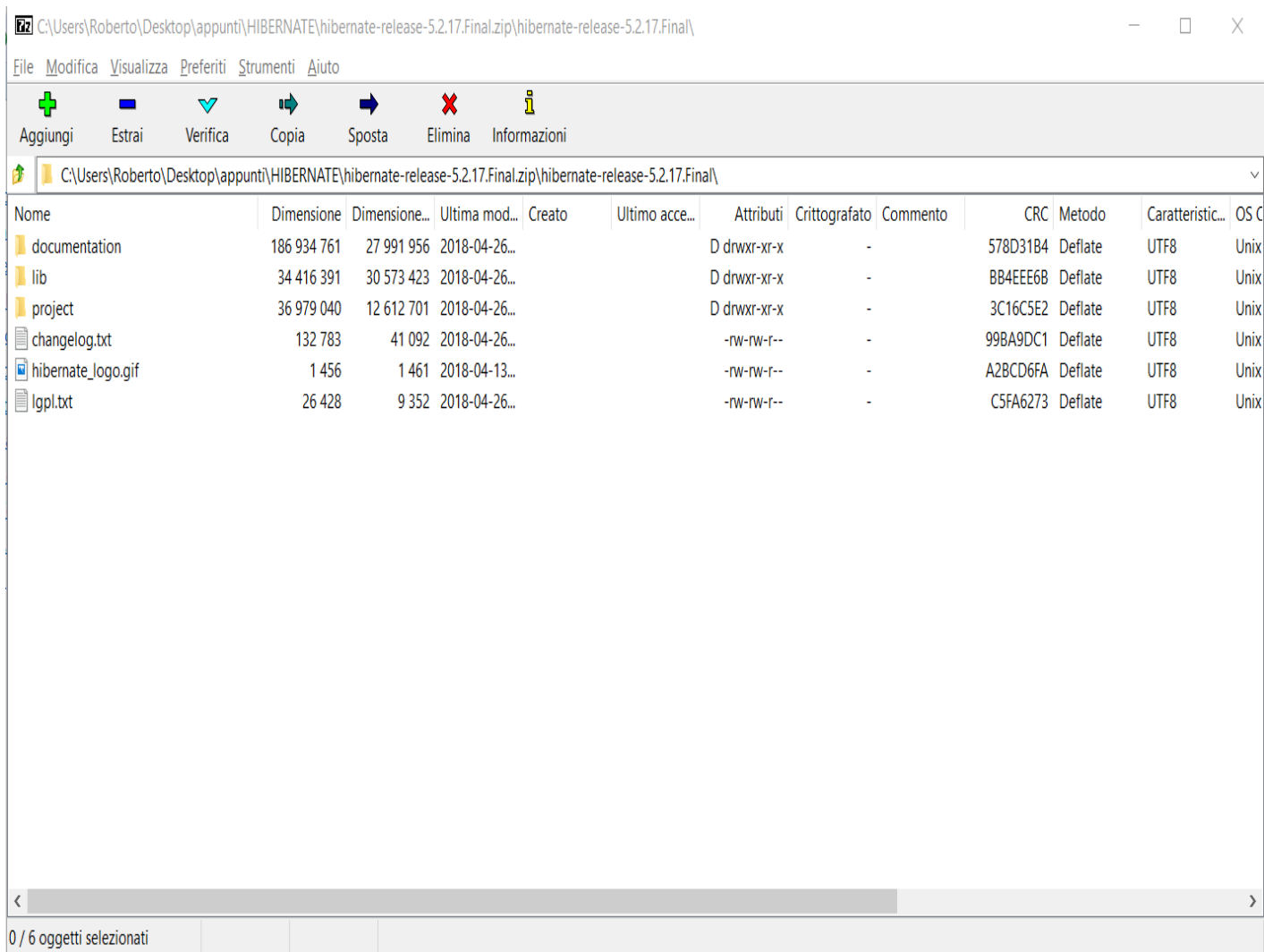
- Files JAR (Java Archive)
- Documentazione
- Codice sorgente
- Altro....

Il link seguente è il link dove scaricare il software Hibernate nella versione 5.2.17 :

→ <https://sourceforge.net/projects/hibernate/files/hibernate-orm/5.2.17.Final/>

Struttura di Hibernate

Una volta scaricato il pacchetto (nel mio caso .zip) apriamolo con un programma come 7zip o winZip, ecc... e vedremo un insieme di cartelle come la seguente immagine:



La cartella più importante di tutte è la cartella denominata lib.

Essa infatti contiene tutti i moduli del software Hibernate.

Facciamo doppio click sulla cartella lib che contiene tutti i moduli/ artefatti (archivi jar) del software Hibernate:

C:\Users\Roberto\Desktop\appunti\HIBERNATE\hibernate-release-5.2.17.Final.zip\hibernate-release-5.2.17.Final\lib\

File Modifica Visualizza Preferiti Strumenti Aiuto

Aggiungi Estrai Verifica Copia Sposta Elimina Informazioni

C:\Users\Roberto\Desktop\appunti\HIBERNATE\hibernate-release-5.2.17.Final.zip\hibernate-release-5.2.17.Final\lib\

Nome	Dimensione	Dimensione...	Ultima mod...	Creato	Ultimo acce...	Attributi	Crittografato	Commento	CRC	Metodo	Caratteristic...	OS C
envers	483 399	426 134	2018-04-26...			D drwxr-xr-x	-		9168E1B7	Deflate	UTF8	Unix
jpa-metamodel-generator	180 574	162 780	2018-04-26...			D drwxr-xr-x	-		2C350F4C	Deflate	UTF8	Unix
optional	18 270 899	16 051 921	2018-04-26...			D drwxr-xr-x	-		C9E6594B	Deflate	UTF8	Unix
osgi	3 962 300	3 473 072	2018-04-26...			D drwxr-xr-x	-		CE836BD9	Deflate	UTF8	Unix
required	8 771 856	7 913 760	2018-04-26...			D drwxr-xr-x	-		E9EDB554	Deflate	UTF8	Unix
spatial	2 747 363	2 545 756	2018-04-26...			D drwxr-xr-x	-		7B5982F0	Deflate	UTF8	Unix

0 / 6 oggetti selezionati

Dentro la cartella **required** abbiamo tutti i moduli necessari (di cui non possiamo fare a meno) per utilizzare Hibernate.

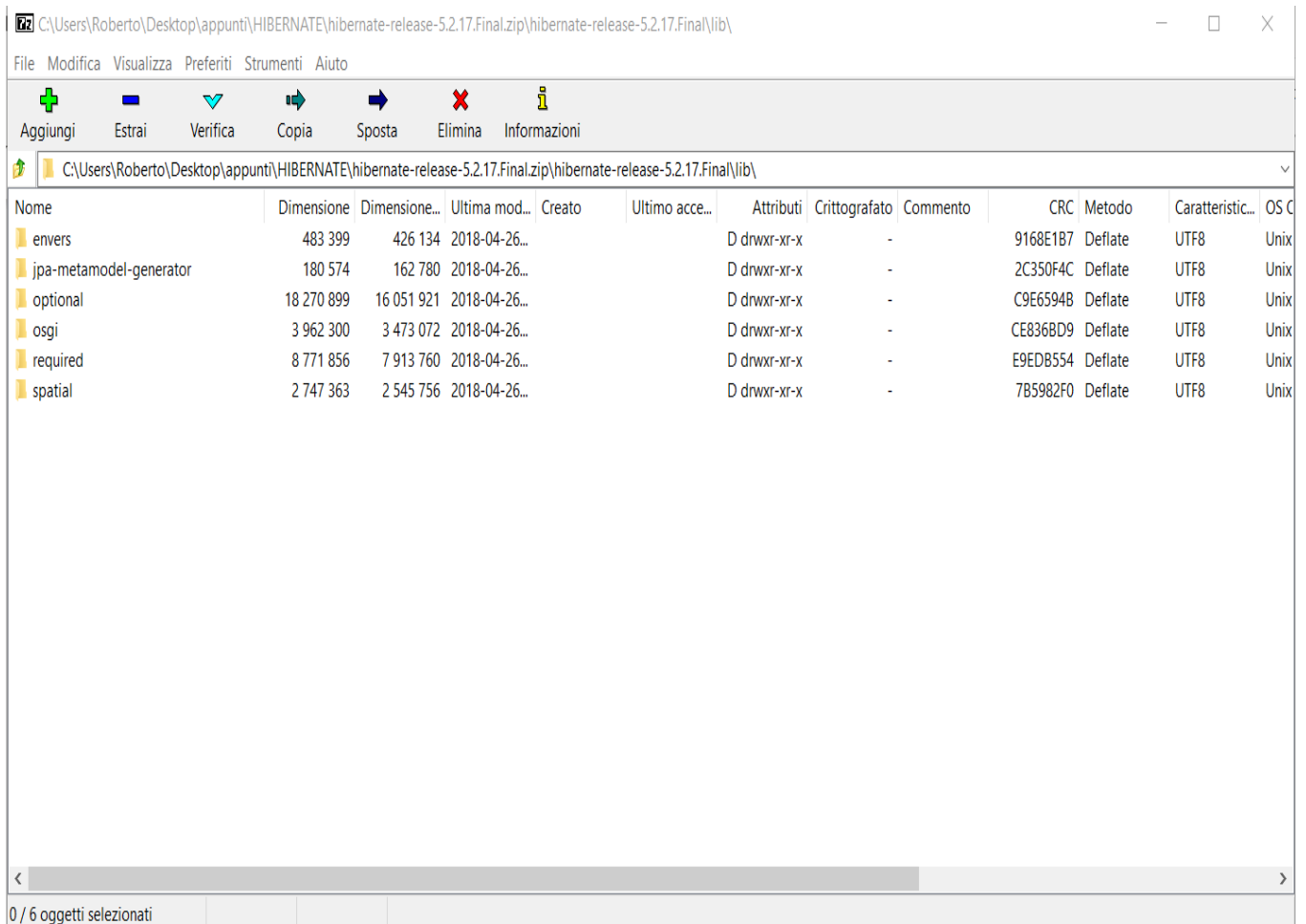
Sicuramente dentro la cartella **required** dovrà esserci il modulo **hibernate-core** di Hibernate essendo il modulo principale (il cuore del software Hibernate).

Quindi facciamo doppio click sulla cartella **required** per visualizzare il contenuto:

C:\Users\Roberto\Desktop\appunti\HIBERNATE\hibernate-release-5.2.17.Final.zip\hibernate-release-5.2.17.Final\lib\required\												
File Modifica Visualizza Preferiti Strumenti Aiuto												
Aggiungi Estrai Verifica Copia Sposta Elimina Informazioni												
C:\Users\Roberto\Desktop\appunti\HIBERNATE\hibernate-release-5.2.17.Final.zip\hibernate-release-5.2.17.Final\lib\required\												
Nome	Dimensione	Dimensione...	Ultima mod...	Creato	Ultimo acce...	Attributi	Crittografato	Commento	CRC	Metodo	Caratteristic...	OS C
antlr-2.7.7.jar	445 288	421 452	2018-04-05...			-rw-rw-r--	-		3684C6C5	Deflate	UTF8	Unix
classmate-1.3.0.jar	64 110	57 363	2018-04-05...			-rw-rw-r--	-		D3C106E0	Deflate	UTF8	Unix
dom4j-1.6.1.jar	313 898	291 488	2018-04-05...			-rw-rw-r--	-		7A3D526A	Deflate	UTF8	Unix
hibernate-commons-annotations-5.0...	75 288	62 597	2018-04-06...			-rw-rw-r--	-		305A0F65	Deflate	UTF8	Unix
hibernate-core-5.2.17.Final.jar	6 739 203	6 046 957	2018-04-26...			-rw-rw-r--	-		A8F41D97	Deflate	UTF8	Unix
hibernate-jpa-2.1-api-1.0.0.Final.jar	113 371	89 096	2018-04-06...			-rw-rw-r--	-		FE4FD89C	Deflate	UTF8	Unix
jandex-2.0.3.Final.jar	186 741	173 620	2018-04-05...			-rw-rw-r--	-		A865811C	Deflate	UTF8	Unix
javassist-3.22.0-GA.jar	739 582	688 085	2018-04-05...			-rw-rw-r--	-		A97CB67E	Deflate	UTF8	Unix
jboss-logging-3.3.1.Final.jar	66 023	59 040	2018-04-05...			-rw-rw-r--	-		C17AD335	Deflate	UTF8	Unix
jboss-transaction-api_1.2_spec-1.0.1...	28 352	24 062	2018-04-05...			-rw-rw-r--	-		7A6F84DE	Deflate	UTF8	Unix
<div> <div><</div> <div>1 / 10 oggetti selezionati</div> <div>6 739 203</div> <div>6 739 203</div> <div>2018-04-26 11:06:06</div> <div>></div> </div>												

Proprio come pensavamo, la cartella required contiene il modulo hibernate-core e in più altri moduli che sono delle dipendenze necessarie di questo.

Vediamo adesso di spiegare meglio è più nel dettaglio il contenuto della cartella lib del file .zip o tar.gz di rilascio per il software Hibernate.



Nome	Dimensione	Dimensione...	Ultima mod...	Creato	Ultimo acce...	Attributi	Crittografato	Commento	CRC	Metodo	Caratteristic...	OS C
envers	483 399	426 134	2018-04-26...			D drwxr-xr-x	-		9168E1B7	Deflate	UTF8	Unix
jpa-metamodel-generator	180 574	162 780	2018-04-26...			D drwxr-xr-x	-		2C350F4C	Deflate	UTF8	Unix
optional	18 270 899	16 051 921	2018-04-26...			D drwxr-xr-x	-		C9E6594B	Deflate	UTF8	Unix
osgi	3 962 300	3 473 072	2018-04-26...			D drwxr-xr-x	-		CE836BD9	Deflate	UTF8	Unix
required	8 771 856	7 913 760	2018-04-26...			D drwxr-xr-x	-		E9EDB554	Deflate	UTF8	Unix
spatial	2 747 363	2 545 756	2018-04-26...			D drwxr-xr-x	-		7B5982F0	Deflate	UTF8	Unix

- La cartella required contiene il jar hibernate-core e tutte le sue dipendenze.

NOTA BENE: Tutti questi jar devono essere inclusi el CLASSPATH (o percorso di classe) indipendentemente dalle funzioni di Hibernate in uso.

- La cartella envers contiene il jar hibernite-envers (oltre alle dipendenze della cartella required e della cartella jpa-metamodel-generator)
- La cartella spatial contiene il jar hiberate-spatial e altri jar che sono sue dipendenze (oltre alle dipendenze della cartella required)

- La cartella osgi contiene il jar hibernate osgi e altri jar che sono sue dipendenze (oltre alle dipendenze della cartella required e jpa-metamodel-generator)
- La cartella jpa-metamodel-generator contiene il jar necessario per generare il metamodel sicuro di tipo API Criteria.
- La cartella optional contiene i jar necessari per i vari pool di connessioni e integrazioni di cache di secondo livello fornite da Hibernate, insieme alle relative dipendenze.

Tutorial sull'utilizzo delle API native di Hibernate e sull'utilizzo del file di mapping hbm.xml

Notiamo che hbm sta per HiBernateMapping.

Obiettivi di questo tutorial sono:

- ➔ Classe di Hibernate: SessionFactory
- ➔ Utilizzare i file hbm.xml di mappatura per hibernate per fornire informazioni sulle classi da mappare
- ➔ Utilizzare le API native di Hibernate

Il File di configurazione di Hibernate

Il file di configurazione di Hibernate è un file che è chiamato "hibernate.cfg" e definisce le informazioni di configurazioni per Hibernate.

Il file hibernate.cfg è un file XML e quindi a estensione .xml. Questo file serve ad Hibernate per poter avere informazioni sul:

- Driver di connessione
- Url di connessione al database
- Username e password dell'utente del database
- e altre informazioni

Ognuna di queste informazioni va specificata all'interno di un elemento property, ovvero un tag chiamato property.

Esempio di 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.dialect">
      org.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>

    <property name="hibernate.connection.url">
      jdbc:mysql://localhost/DBSchemaName
    </property>
    <property name="hibernate.connection.username">
      testUserName
    </property>
    <property name="hibernate.connection.password">
      testPassword
    </property>

    <!-- List of XML mapping files -->
    <mapping resource="HibernatePractice/Employee.hbm.xml"/>

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

```

<?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">

```

<-- PROLOGO

```

<hibernate-configuration> <-- ROOT ELEMENT
  <session-factory>
    <property name="hibernate.dialect"> <-- PROPERTY ELEMENT
      org.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.connection.driver_class"> <-- PROPERTY ELEMENT
      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.url"> <-- PROPERTY ELEMENT
      jdbc:mysql://localhost/DBSchemaName
    </property>
    <property name="hibernate.connection.username"> <-- PROPERTY ELEMENT
      testUserName
    </property>
    <property name="hibernate.connection.password"> <-- PROPERTY ELEMENT
      testPassword
    </property>

    <!-- List of XML mapping files -->
    <mapping resource="HibernatePractice/Employee.hbm.xml"/>

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

```

Il PROLOGO è una parte opzionale del file XML (anche se fortemente consigliato metterla) che specifica:

1. la dichiarazione XML
2. la dichiarazione del DDT

Ovvero quel file che permette di validare il file XML in modo da assicurare che il file XML sia bene formato.

Nel nostro caso abbiamo come dichiarazione del DDT il link dove si trova fisicamente il DDT per questo file XML.

Notiamo alcune cose all'interno del file XML:

1. Ogni tag di apertura ha un corrispettivo tag di chiusura.
Ad eccezione di alcuni tag che sono "autochiudenti" tipo il tag con nome mapping.
2. Una coppia di tag : una di apertura e una di chiusura con lo stesso nome è chiamato element.

Nel file di prima abbiamo 8 element.

3. Il tag che incorpora tutti chiamato <hibernate-configuration> che abbiamo colorato in verde è il root element, ovvero il tag radice di tutti.
4. Nel file precedente abbiamo definito 4 property, ognuno specifica una diversa informazione per Hibernate come ad esempio:
 - i. Nome del driver
 - ii. URL di connessione
 - iii. Username dell'utente
 - iv. Password dell'utente
 - v. Dialecto sql da usare

Il tag "autochiudente" mapping serve per dare delle informazioni ad Hibernate su dove si trova il file di mapping (hbm.xml) che contiene informazioni sulla classe da mappare in tabella.

Esempio:

```
<?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> <-- ROOT ELEMENT
  <session-factory>
    <property name="hibernate.dialect">
      org.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>

    <property name="hibernate.connection.url">
      jdbc:mysql://localhost/DBSchemaName
    </property>
    <property name="hibernate.connection.username">
      testUserName
    </property>
    <property name="hibernate.connection.password">
      testPassword
    </property>

    <!-- List of XML mapping files -->
    <mapping resource="HibernatePractice/Employee.hbm.xml"/> <-- MAPPING ELEMENT
  </session-factory>
</hibernate-configuration>
```

Notiamo che il tag (o element) mapping contiene un attributo di nome resource dove specifichiamo il percorso completo o assoluto di dove trovare il file Employer.hbm.xml (che contiene informazioni sul mapping della classe in tabella).

La classe java Entity

La classe Entity per questa esercitazione è la classe User.

Note sulle Entity:

1. Una classe Entity non è altro che un JavaBean.
Ovvero una classe che ha:
 - Variabili di istanze private (incapsulamento)
 - Costruttore senza argomenti
 - Metodi get e set
 - Implementa l'interfaccia Serializable
2. Il costruttore senza argomenti è una convenzione JavaBean ma anche un requisito essenziale per tutte le classi che devono essere persistite.
Infatti il costruttore senza argomenti viene usato da Hibernate per serializzare e deserializzare l'oggetto sul database, se esso non è presente viene sollevata una eccezione a runtime da Hibernate.

Il file di mappatura hbm.xml

Il file di mapping (o mappatura) per questa esercitazione è il file User.hbm.xml.

Notiamo che anche questo file è un file XML in quanto ha estensione .xml.

Questo file xml contiene informazione (che verranno usate da hibernate) sul mappaggio della classe Entity denominata User in una tabella del database.

Hibernate utilizza queste informazioni (metadati) del file hbm.xml per determinare:

1. Come caricare oggetti della classe persistente a partire dai record della corrispondente tabella del database
2. Come memorizzare (persistere) oggetti della classe persistente in record della corrispondente tabella del database

Il file di di mapping hbm.xml è una delle possibili scelte per fornire ad hibernate queste informazioni. Vedremo più avanti che esiste un modo più innovativo di farlo che sostituisce la creazione del file hbm.xml e che consiste nell'annotare la classe da mappare con dei particolari tag chiamati annotation.

Esempio di file hbm.xml:

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

<hibernate-mapping>
  <class name = "Employee" table = "EMPLOYEE">

    <meta attribute = "class-description">
      This class contains the employee detail.
    </meta>

    <id name = "id" type = "int" column = "id">
      <generator class="native"/>
    </id>

    <property name = "firstName" column = "first_name" type = "string"/>
    <property name = "lastName" column = "last_name" type = "string"/>
    <property name = "salary" column = "salary" type = "int"/>

  </class>
</hibernate-mapping>
```

Questo file xml specifica il mapping di una sola classe Entity (infatti abbiamo un solo tag class) di nome Employee (attributo del tag class) che contiene 4 variabili di istanza:

- Id
- firstName
- lastName
- salary

Questo file hbm.xml serve ad hibernate per avere tutte le informazioni per il mapping della classe. Queste informazioni possono essere:

- quale classe mappare in tabella del database (ovvero la classe da persistere)
- quali variabili di istanza della classe mappare in colonne della tabella
- come mappare l'id della tabella
- e altre informazioni (come ad esempio le relazioni con altre tabelle)

Cerchiamo adesso di spiegare più nel dettaglio il contenuto di questo file xml:

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

<!-- PROLOGO

```
<hibernate-mapping> <-- ROOT ELEMENT
  <class name = "Employee" table = "EMPLOYEE"> <-- CLASS ELEMENT
    <meta attribute = "class-description"> <-- META ELEMENT
      This class contains the employee detail.
    </meta>
    <id name = "id" type = "int" column = "id"> <-- ID ELEMENT
      <generator class="native"/>
    </id>
    <property name = "firstName" column = "first name" type = "string"/> <-- PROPERTY ELEMENT
    <property name = "lastName" column = "last_name" type = "string"/> <-- PROPERTY ELEMENT
    <property name = "salary" column = "salary" type = "int"/> <-- PROPERTY ELEMENT
  </class>
</hibernate-mapping>
```

Possiamo notare che all'inizio del file xml è presente il PROLOGO che abbiamo spiegato già più sopra.

Le cose che possiamo notare all'interno del file XML sono:

1. Il tag che incorpora tutti chiamato <hibernate-mapping> che abbiamo colorato in verde è il root element, ovvero il tag radice di tutti.
2. Il tag class specifica che deve essere mappata una classe in tabella del database. La classe che deve essere mappata è specificata all'interno

dell'attributo name del tag class. Inoltre è possibile anche specificare mediante l'attributo table il nome della tabella in cui mappare questa classe.

NOTA BENE: se non è presente l'attributo name del tag class allora per default il nome della tabella del database avrà lo stesso nome dell'oggetto da mappare.

3. Il tag id serve per mappare la chiave primaria della tabella.
In questo caso file viene specificato di mappare la variabile di istanza id nella colonna id della tabella.
4. Il tag property serve per mappare le variabili di istanza della classe in colonne normali (non chiave primaria (indice)) della tabella

BIBLIOGRAFIA

- 1 https://docs.jboss.org/hibernate/orm/5.2/quickstart/html_single/#obtaining
- 2 https://www.tutorialspoint.com/hibernate/hibernate_examples.htm
- 3 https://docs.jboss.org/hibernate/orm/5.4/quickstart/html_single/#preface
- 4 <https://www.filetypeadvisor.com/it/extension/tgz>