Association bidirectionnelle
balise inverse

Mapping Theme

```
<hibernate-mapping  package="com.plb.etechno.j12.exemple.metier">
  <class name="Theme" table="TTheme">
      <!– id et autres propriétés →

      <set name="motclefs"  inverse=" true ">
        <key column="IDTheme"/>
        <one-to-many class="MotClef"/>
      </set>
   </class>
</hibernate-mapping>
```

Code

```
public class Theme {
     ……..
          public void addMotClef(MotClef mot){
          motclefs.add(mot);
          mot.setTheme(this);
     }
```

Association OneToOne bidirectionnelle

```
<class name="Person">
  <id name="id" column="personId">
    <generator class="native"/>
  </id>
  <many-to-one name="address" column="addressId"
    unique="true" not-null="true"/>
</class>
<class name="Address">
  <id name="id" column="addressId">
    <generator class="native"/>
  </id>
  <one-to-one name="person" property-ref="address"/>
</class>
```

Association OneToMany bidirectionnelle

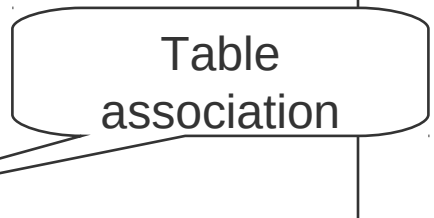
```
<class name="Theme">
  <id name="id" column="id">
    <generator class="native"/>
  </id>
  <set name="mots" inverse="true">
    <key column="IDTheme"/>
    <one-to-many class="MotClef"/>
  </set>
</class>
<class name="MotClef">
  <id name="id" column="id">
    <generator class="native"/>
  </id>
  <many-to-one name="address" column="IDTheme" not-null="true"/>
</class>
```

Association many-to-many
Le mapping

```
<hibernate-mapping  package="com.tsystems.etechno.j12.exemple.metier">
 <class name="Item" table="TItem" discriminator-value="IT">
     <id  name="id" column="ID">
       <generator class="native"/>
     </id>
     <!– autres propriétés →

     <set name="motclefs" table="titem_motclef" >
       <key column="IDItem"/>
       <many-to-many class="MotClef" column="IDMotClef"/>
     </set>

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



Collections

Les *Collection*, *List*, *Map* et *Set* peuvent être mappées via les annotations *@OneToMany* ou *@ManyToMany*

Si on utilise *hbm*, il faut utiliser la balise adéquate (<*set*>, <*list*>, <*map*>, <*bag*>, <*array*>, <*primitive-array*>)

Lorsque la collection est persistée ou chargée, elle est remplacée par une collection de type Hibernate qui respecte l'interface initiale

Collections Hibernate

```
Cat cat = new DomesticCat();
Cat kitten = new DomesticCat();
....
Set kittens = new HashSet();
kittens.add(kitten);
cat.setKittens(kittens);
session.persist(cat);
kittens = cat.getKittens(); // Okay,
(HashSet) cat.getKittens(); // Error!
```

Mapping de Collection
<set> et <bag>

```
<set name="motclefs" >
        <key column="IDTheme"/>
        <one-to-many class="MotClef"/>
 </set>
```

Set motclefs = new HashSet<MotClef>();

[MotClef@76, MotClef@43]

{ Non ordonné a priori }

```
<bag name="motclefs">
        <key column="IDTheme"/>
        <one-to-many class="MotClef"/>
 </bag>
```

List motclefs = new ArrayList<MotClef>();

[MotClef@43, MotClef@76]

{ Ordonné suivant le chargement }

Mapping de Collection
<list> et <map>

Tag Hibernate

```
<list name="motclefs" >
        <key column="Idtheme"/>
        <list-index column="id"/>
        <one-to-many class="MotClef"/>
</list>
```

Collection Java

```
List motclefs = new ArrayList<MotClef>();
```

[null,null,…, null,fantôme, gothique]

```
<map name="motclefs">
    <key column="IDTheme"/>
    <map-key column="mot_clef"
                type="string« />
     <one-to-many class="MotClef"/>
 </map>
```

```
Map<String,MotClef> motclefs =
        new HashMap<String,MotClef>();
```

[{« fantome », Motclef@43},
{« gothique »,MotClef@76}]

Mapping des associations