**Hibernate – Load or Get Entity Example**

In previous tutorials, we learned about [**connecting pooling**](https://howtodoinjava.com/hibernate/hibernate-c3p0-connection-pool-configuration-tutorial/), [**persistence Life-cycle States**](https://howtodoinjava.com/hibernate/hibernate-entity-persistence-lifecycle-states/), [**association Mappings**](https://howtodoinjava.com/hibernate/how-to-define-association-mappings-between-hibernate-entities/) and [**entities Equality and Identity concepts**](https://howtodoinjava.com/hibernate/hibernate-entities-equality-and-identity/). In next some tutorials, I am going to cover some concepts around **CRUD operations** which you will do while interacting with database though [**hibernate**](https://howtodoinjava.com/hibernate-tutorials/). Let’s start the discussion with loading an hibernate entity using either load or get method.

**1) Loading an hibernate entity using session.load() method**

Hibernate’s Session interface provides several load() methods for loading entities from your database. Each load() method requires the object’s primary key as an identifier, and it is mandatory to provide it. In addition to the ID, Hibernate also needs to know which class or entity name to use to find the object with that ID. After the load() method returns, you need to cast the returned object to suitable type of class to further use it. It’s all what load() method need from you to work it correctly.

Let’s look at different flavors of load() method available in hibernate session:

1. public Object load(Class theClass, Serializable id) throws HibernateException
2. public Object load(String entityName, Serializable id) throws HibernateException
3. public void load(Object object, Serializable id) throws HibernateException

First load() method need the class type which you would like to load along with unique ID. Second load() method asks for **entityName** directly and unique ID. Both method return the populated entity object as return value which you will cast to desired type.

The last load() method takes an object as an argument. The object should be of the same class as the object you would like loaded, and it should be empty. Hibernate will populate that object with the object you requested.

The other load() methods available through hibernate session take a lock mode as an argument too. The lock mode specifies whether Hibernate should look into the cache for the object and which database lock level Hibernate should use for the row (or rows) of data that represent this object. In official documentation, hibernate developers claim that *Hibernate will usually pick the correct lock mode* for you, although in some situations it is important to manually choose the correct lock.

We will discuss more about locks when we will be done with basic hibernate concepts.

Let’s look at the examples of each load method in simplest form to be clear what we read above.

|  |
| --- |
| public class TestHibernate  {     public static void main(String[] args)     {        Session sessionOne = HibernateUtil.getSessionFactory().openSession();        sessionOne.beginTransaction();          // Create new Employee object        EmployeeEntity emp = new EmployeeEntity();        emp.setFirstName("Lokesh");        emp.setLastName("Gupta");          //Save employee        sessionOne.save(emp);        //store the employee id generated for future use        Integer empId = emp.getEmployeeId();        sessionOne.getTransaction().commit();          /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/          //Let's open a new session to test load() methods        Session sessionTwo = HibernateUtil.getSessionFactory().openSession();        sessionTwo.beginTransaction();          //first load() method example        EmployeeEntity emp1 = (EmployeeEntity) sessionTwo.load(EmployeeEntity.class, empId);        System.out.println(emp1.getFirstName() + " - " +emp1.getLastName());          //Let's verify the entity name        System.out.println(sessionTwo.getEntityName(emp1));          sessionTwo.getTransaction().commit();          /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/          Session sessionThree = HibernateUtil.getSessionFactory().openSession();        sessionThree.beginTransaction();          //second load() method example        EmployeeEntity emp2 = (EmployeeEntity) sessionThree.load("com.howtodoinjava.demo.entity.EmployeeEntity", empId);        System.out.println(emp2.getFirstName() + " - " +emp2.getLastName());          sessionThree.getTransaction().commit();          /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/          Session sessionFour = HibernateUtil.getSessionFactory().openSession();        sessionFour.beginTransaction();          //third load() method example        EmployeeEntity emp3 = new EmployeeEntity();        sessionFour.load(emp3 , empId);        System.out.println(emp3.getFirstName() + " - " +emp3.getLastName());          sessionFour.getTransaction().commit();          HibernateUtil.shutdown();     }  }      Output:    Hibernate: insert into Employee (FIRST\_NAME, LAST\_NAME, ID) values (?, ?, ?)  Hibernate: select employeeen0\_.ID as ID1\_1\_0\_, employeeen0\_.FIRST\_NAME as FIRST\_NA2\_1\_0\_, employeeen0\_.LAST\_NAME as LAST\_NAM3\_1\_0\_              from Employee employeeen0\_ where employeeen0\_.ID=?  Lokesh - Gupta      //First load method  com.howtodoinjava.demo.entity.EmployeeEntity  Hibernate: select employeeen0\_.ID as ID1\_1\_0\_, employeeen0\_.FIRST\_NAME as FIRST\_NA2\_1\_0\_, employeeen0\_.LAST\_NAME as LAST\_NAM3\_1\_0\_              from Employee employeeen0\_ where employeeen0\_.ID=?  Lokesh - Gupta      //Second load method  Hibernate: select employeeen0\_.ID as ID1\_1\_0\_, employeeen0\_.FIRST\_NAME as FIRST\_NA2\_1\_0\_, employeeen0\_.LAST\_NAME as LAST\_NAM3\_1\_0\_              from Employee employeeen0\_ where employeeen0\_.ID=?  Lokesh - Gupta      //Third load method |

So we are able to load the entity from all three load methods successfully. Now move on to get()method.

**2) Loading an hibernate entity using session.get() method**

The get() method is very much similar to load() method. The get() methods take an identifier and either an entity name or a class. There are also two get() methods that take a lock mode as an argument, but we will discuss lock modes later. The rest get() methods are as follows:

1. public Object get(Class clazz, Serializable id) throws HibernateException
2. public Object get(String entityName, Serializable id) throws HibernateException

There is not much difference in code while working with either load() or get() method, all you need is to replace the load() method with get() method in first two examples. There is no get() equivalent to last load() method.

You can modify above example and test the code. Let me know if you find any problem.

**Difference between load() and get() method in hibernate session**

The difference between both methods lies in return value “if the identifier does not exist in database”. **In case of get() method you will get return value as NULL if identifier is absent; But in case of load() method, you will get a runtime exception** something like below:

|  |
| --- |
| Exception in thread "main" org.hibernate.ObjectNotFoundException: No row with the given identifier exists:  [com.howtodoinjava.demo.entity.EmployeeEntity#23]  at org.hibernate.internal.SessionFactoryImpl$1$1.handleEntityNotFound(SessionFactoryImpl.java:253)  at org.hibernate.event.internal.DefaultLoadEventListener.load(DefaultLoadEventListener.java:219)  at org.hibernate.event.internal.DefaultLoadEventListener.proxyOrLoad(DefaultLoadEventListener.java:275)  at org.hibernate.event.internal.DefaultLoadEventListener.onLoad(DefaultLoadEventListener.java:151)  at org.hibernate.internal.SessionImpl.fireLoad(SessionImpl.java:1070)      at org.hibernate.internal.SessionImpl.load(SessionImpl.java:940) |

That’s all for this beginner level tutorial discussing**loading hibernate entities**. In coming pots, we will discuss more concepts around **CRUD operations**.