HIBERNATE 4 EXAMPLE PROGRAMS

USING NETBEANS IDE

Hibernate 4 Example Programs

L. Mohan Arun

This book is for sale at http://leanpub.com/hibernate4-example-programs

This version was published on 2013-11-25

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Also By L. Mohan Arun

Winning PPC ads Blueprint

ActiveMQ brief for learners

Making the Most of NetBeans

Contents

1 Basic CRUD using mapping files

Create the database: Use the MySQL command line client:

mysql -u -root -p

Enter your password and log in.

Use this command to create database.

CREATE DATABASE mybusiness;

USE mybusiness;

Type in this table creation script.

MySQL Table creation script

```
CREATE TABLE customers(
C_ID int NOT NULL AUTO_INCREMENT,
name varchar(20),
address  varchar(20),
PRIMARY KEY(C_ID)
);
```

This is the Hibernate configuration file:

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DT\
D 3.0//EN"
"http://www.hibernate.org/dtd/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</propert\
y>
<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mybusiness<\
/property>
```

```
<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">power0l</property>
<property name="hibernate.show_sql">true</property>
<mapping resource="customersmapping.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

This is the Hibernate mappping file:

customersmapping.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN" "\
http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="hibernatetest.Customer" table="customers">
<id column="C_ID" name="customerID" type="int">
<generator class="native">
</generator></id>
<property name="customerName">
<column name="name">
</column></property>
<property name="customerAddress">
<column name="address">
</column></property>
</class>
</hibernate-mapping>
```

This below is the POJO class Customer.java that maps with the customers table in the database.

This POJO class is common to the Create, Read, Update, Delete programs below (place it in the same project)

Customer.java

```java
package hibernatetest;

public class Customer {

private int customerID;
private String customerName;
private String customerAddress;

public Customer() {}

public String getCustomerAddress() {
return customerAddress;
}

public void setCustomerAddress(String customerAddress) {
this.customerAddress = customerAddress;
}

public int getCustomerID() {
return customerID;
}

public void setCustomerID(int customerID) {
this.customerID = customerID;
}

public String getCustomerName() {
return customerName;
}

public void setCustomerName(String customerName) {
this.customerName = customerName;
}

}
```

1.1 Create

This program intends to demonstrate how to add a new customer record to the customers table.

CreateTest.java

```
package hibernatetest;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;

public class CreateTest {

public static void main(String[] args) {

Session session = null;
try {

Configuration configuration = new Configuration();
configuration.configure();

ServiceRegistry serviceRegistry = new ServiceRegistryBuilder().applySettings(conf\
iguration.getProperties()).buildServiceRegistry();
SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry\
);
session = sessionFactory.openSession();

session.beginTransaction();

System.out.println("Adding a customer record !");

Customer customer = new Customer();
customer.setCustomerName("Customer-a");
customer.setCustomerAddress("Address1");

session.save(customer);
session.getTransaction().commit();

System.out.println("Done!");
```

```
} catch(Exception e) {
System.out.println(e.getMessage());
}

finally {
session.flush();
session.close();
}

}
}
```

1.2 Read

This program intends to demonstrate how to list the records of the customers table.

ReadTest.java

```
package hibernatetest;

import java.util.Iterator;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;

public class ReadTest {

public static void main(String[] args) {

Session session = null;
try {
Configuration configuration = new Configuration();
configuration.configure();

ServiceRegistry serviceRegistry = new ServiceRegistryBuilder().applySettings(conf\
iguration.getProperties()).buildServiceRegistry();
SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry\
);

session = sessionFactory.openSession();
session.beginTransaction();

String HQL_QUERY ="from Customer customers";
org.hibernate.Query query = session.createQuery(HQL_QUERY);
System.out.println("Reading values...");

for (Iterator it=query.iterate(); it.hasNext(); ) {
        Customer customer = (Customer) it.next();
        System.out.println("----------");
        System.out.println("Name:" + customer.getCustomerName());
        System.out.println("Address:"+customer.getCustomerAddress());
        System.out.println("ID:"+customer.getCustomerID());
```

```
        System.out.println("----------");
}
session.getTransaction().commit();
System.out.println("Done!");
}

catch(Exception e){
System.out.println(e.getMessage());
}

finally{
session.flush();
session.close();
}
}
}
```

This program below is very similar to the previous ReadTest.java, the only difference being, it demonstrates the use of query.list() (list method in query interface) which is similar to the use of query.iterate().

ReadTest2.java

```
package hibernatetest;

import java.util.List;
import java.util.Iterator;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;

/**
 *
 * @author Administrator
 */
public class ReadTest2 {

/**
 * @param args the command line arguments
```

```
 */
public static void main(String[] args) {
        Session session = null;
try{
        Configuration configuration = new Configuration();
        configuration.configure();

        ServiceRegistry serviceRegistry = new ServiceRegistryBuilder().applySettings(con\
figuration.getProperties()).buildServiceRegistry();
        SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistr\
y);

        session = sessionFactory.openSession();
        session.beginTransaction();
        System.out.println(org.hibernate.Version.getVersionString());
        String HQL_QUERY ="from Customer customers";
        org.hibernate.Query query = session.createQuery(HQL_QUERY);

        List list = query.list();

        System.out.println("Retrieving " + list.size() + " Records:");
         for(Iterator it=list.iterator();it.hasNext();){
                 Customer customer = (Customer) it.next();
                 System.out.println("Name:" + customer.getCustomerName());
                 System.out.println("Address:"+customer.getCustomerAddress());
                 System.out.println("ID:"+customer.getCustomerID());
                 System.out.println("----------");
                }
        session.getTransaction().commit();
        System.out.println("Done!");
        }

catch(Exception e){
        System.out.println(e.getMessage());
}
finally{
session.flush();
session.close();
}
}
}
```

1.3 Update

This program intends to demonstrate how to update a record in the customers table.

UpdateTest.java

```
package hibernatetest;

import java.util.Iterator;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;

public class UpdateTest {

public static void main(String[] args) {
Session session = null;

try {
Configuration configuration = new Configuration();
configuration.configure();

ServiceRegistry serviceRegistry = new ServiceRegistryBuilder().applySettings(conf\
iguration.getProperties()).buildServiceRegistry();
SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry\
);

session = sessionFactory.openSession();
session.beginTransaction();

String HQL_QUERY ="from Customer customers where customers.customerID = :customer\
Id";
org.hibernate.Query query = session.createQuery(HQL_QUERY);
//Prepared statement
query.setParameter("customerId",1);

for (Iterator it = query.iterate(); it.hasNext(); ) {
Customer customer = (Customer) it.next();
customer.setCustomerName("Customer1");
}
```

```
session.getTransaction().commit();
System.out.println("Done!");

} catch(Exception e) {
System.out.println(e.getMessage());
}

finally {
session.flush();
session.close();
}

}
}
```

1.4 Delete

This program sets out to demonstrate how to delete a record from the customers table.

DeleteTest.java

```
package hibernatetest;

import java.util.Iterator;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;

public class DeleteTest {

public static void main(String[] args) {

Session session = null;

try {
Configuration configuration = new Configuration();
configuration.configure();

ServiceRegistry serviceRegistry = new ServiceRegistryBuilder().applySettings(conf\
iguration.getProperties()).buildServiceRegistry();
SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry\
);

session = sessionFactory.openSession();
session.beginTransaction();

Customer cust = new Customer();
cust.setCustomerID(2);
session.delete(cust);
session.getTransaction().commit();

System.out.println("Done!");
}

catch(Exception e) {
```

```
System.out.println(e.getMessage());
}

finally {
session.flush();
session.close();
}
}
}
```