Index.html

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**h3**>Hibernate Annoted Query Demo</**h3**>

<**br**>

<**a** href=*"HibernateQueryDemo"*>Hibernate Query Demo</**a**><**br**>

</**body**>

</**html**>

InitDemo.java

package com.simpli;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

@WebServlet("/init")

public class InitDemo extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

PrintWriter out = response.getWriter();

out.print("<html><body>");

///STEP 1

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

out.println("Hibernate Session opened. <br>");

session.close();

out.println("Hibernate Session closed. <br>");

//STEP 2 execute HQL command

//for know we will only test if the connection is established with MYSQL server

out.print("<body><html>");

}

}

HibernateUtil.java

package com.simpli;

import org.hibernate.SessionFactory;

import org.hibernate.boot.Metadata;

import org.hibernate.boot.MetadataSources;

import org.hibernate.boot.registry.StandardServiceRegistry;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {

private static final SessionFactory sessionFactory;

static {

try {

StandardServiceRegistry standardRegistry = new StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = new MetadataSources(standardRegistry).getMetadataBuilder().build();

sessionFactory = metaData.getSessionFactoryBuilder().build();

} catch (Throwable th) {

throw new ExceptionInInitializerError(th);

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

}

HibernateQuery.java

package com.simpli;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import com.ecommerce.EProduct;

@WebServlet("/HibernateQueryDemo")

public class HibernateQueryDemo extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter out = response.getWriter();

out.print("<html><body>");

///STEP 1

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

out.println("Hibernate Session opened. <br>");

//STEP 2 execute HQL command

//for know we will only test if the connection is established with MYSQL server

List<EProduct>eproduct = session.createQuery("from EProduct").list();

out.println("<br> Data from the eproduct table <table border = 1>");

//prod is the object of eproduct

for(EProduct prod: eproduct) {

out.println("<tr><td>" +prod.getId() +"<td>" +prod.getName()

+ "<td>" +prod.getPrice());

}

session.close();

out.println("Hibernate session closed.<br>");

out.print("<body><html>");

}

}

hibernate.cfg.xml

<?**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.cj.jdbc.Driver</**property**>

<**property** name=*"connection.url"*>jdbc:mysql://localhost:3306/eproduct</**property**>

<**property** name=*"connection.username"*>root</**property**>

<**property** name=*"connection.password"*>8521535902</**property**>

<**mapping** class=*"com.ecommerce.EProduct"*/>

</**session-factory**>

</**hibernate-configuration**>

EProduct.java

package com.ecommerce;

import java.math.BigDecimal;

import javax.persistence.\*;

@Entity

@Table(name="product")

public class EProduct {

@Id

@GeneratedValue

@Column(name="ID")

private int id;

@Column(name="Name")

private String name;

@Column(name="Price")

private BigDecimal price;

public EProduct() {

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public BigDecimal getPrice() {

return price;

}

public void setPrice(BigDecimal price) {

this.price = price;

}

@Override

public String toString() {

return "EProduct [id=" + id + ", name=" + name + ", price=" + price + "]";

}

}

log4j.properties

log4j.rootLogger=DEBUG, Appender1,Appender2

log4j.appender.Appender1=org.apache.log4j.ConsoleAppender

log4j.appender.Appender1.layout=org.apache.log4j.PatternLayout

log4j.appender.Appender1.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n

log4j.appender.Appender2=org.apache.log4j.FileAppender

log4j.appender.Appender2.File=\C:\Users\Shalini Bharti\Downloads\applog.txt

log4j.appender.Appender2.layout=org.apache.log4j.PatternLayout

log4j.appender.Appender2.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n

# Log everything. Good for troubleshooting

log4j.logger.org.hibernate=INFO

# Log all JDBC parameters

log4j.logger.org.hibernate.type=ALL

Output

