**Eproduct.java:**

package services;

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 org.hibernate.cfg.Configuration;

import beans.EProduct;

@WebServlet("/RetreiveServlet")

public class RetreiveServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public RetreiveServlet() {

super();

// TODO Auto-generated constructor stub

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();

Session session = sessionFactory.openSession();

// using HQL

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

session.close();

PrintWriter out = response.getWriter();

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

out.println("<b>Product Listing</b><br>");

for (EProduct p : list) {

out.println("ID: " + String.valueOf(p.getID()) + ", Name: " + p.getName() + ", Price: "

+ String.valueOf(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString() + "<br>");

}

out.println("</body></html>");

} catch (Exception ex) {

throw ex;

}

}

}

**AddServelet.Java:**

package services;

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;

import org.hibernate.cfg.Configuration;

import beans.EProduct;

@WebServlet("/AdderServlet")

public class AdderServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public AdderServlet() {

super();

// TODO Auto-generated constructor stub

}

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

// TODO Auto-generated method stub

response.setContentType("text/html");

PrintWriter out=response.getWriter();

long price=Integer.parseInt(request.getParameter("price"));

String name=request.getParameter("name");

EProduct ep=new EProduct();

ep.setName(name);

ep.setPrice(price);

SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();

Session session = sessionFactory.openSession();

session.beginTransaction();

Long i=(Long)session.save(ep);

session.getTransaction().commit();

session.close();

if (i > 0)

out.println("Record inserted");

else

out.println("Record not inserted");

}

}

**RetrieveServelet.Java :**

package services;

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 org.hibernate.cfg.Configuration;

import beans.EProduct;

@WebServlet("/RetreiveServlet")

public class RetreiveServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public RetreiveServlet() {

super();

// TODO Auto-generated constructor stub

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();

Session session = sessionFactory.openSession();

// using HQL

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

session.close();

PrintWriter out = response.getWriter();

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

out.println("<b>Product Listing</b><br>");

for (EProduct p : list) {

out.println("ID: " + String.valueOf(p.getID()) + ", Name: " + p.getName() + ", Price: "

+ String.valueOf(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString() + "<br>");

}

out.println("</body></html>");

} catch (Exception ex) {

throw ex;

}

}

}

**hibernate.cfg.xml:**

<?xml version='1.0' encoding='utf-8'?>

<!--

~ Hibernate, Relational Persistence for Idiomatic Java

~

~ License: GNU Lesser General Public License (LGPL), version 2.1 or later.

~ See the lgpl.txt file in the root directory or <http://www.gnu.org/licenses/lgpl-2.1.html>.

-->

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory

<property name="format\_sql">false</property>

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

<property name="connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

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

<property name="connection.password">toor</property>

**<!-- JDBC connection pool (use the built-in) -->**

<property name="connection.pool\_size">100</property>

**<!-- SQL dialect -->**

<property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>

**<!-- Disable the second-level cache -->**

<property name="cache.provider\_class">org.hibernate.cache.internal.NoCacheProvider</property>

**<!-- Echo all executed SQL to stdout -->**

<property name="show\_sql">true</property>

**<!-- Drop and re-create the database schema on startup -->**

<property name="hbm2ddl.auto">update</property>

**<!-- <property name="hbm2ddl.auto">update</property> -->**

<mapping class="beans.EProduct"></mapping>

</session-factory>

</hibernate-configuration>

**Web.xml:**

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<servlet-name>RetrieveServlet</servlet-name>

<display-name>RetrieveServlet</display-name>

<description></description>

<servlet-class>services.RetrieveServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>RetrieveServlet</servlet-name>

<url-pattern>/RetrieveServlet</url-pattern>

</servlet-mapping>

</web-app>

**index.jsp:**

<html>

<body>

<form action="AdderServlet" method="get">

Enter Product name:<input type="text" name="name">

Enter Price<input type="number"name="price">

<button>Add</button>

</form>

<a href="RetreiveServlet">Show Table</a>

</body>

</html>

**pom.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.na</groupId>

<artifactId>ProductAdding</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>ProductAdding Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.5.6.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

</dependencies>

<build>

<finalName>ProductAdding</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>