1. **Configure Hibernate in Eclipse IDE.**

**Source code:**

<?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.jdbc.Driver</property>

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

<property name="connection.username">mysql user id</property>

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

</session-factory>

</hibernate-configuration>

**HTML:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Configuration Example</title>

</head>

<body>

<a href="init">Initialize Hibernate</a><br>

</body>

</html>

**Code:**

package com.ecommerce;

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;

}

}

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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 javax.transaction.\*;

import javax.xml.bind.\*;

import java.io.Serializable;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.ecommerce.EProduct;

import com.ecommerce.HibernateUtil;

@WebServlet("/InitHibernate")

public class InitHibernate extends HttpServlet {

private static final long serialVersionUID = 1L;

public InitHibernate() {

super();

// TODO Auto-generated constructor stub

}

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

// TODO Auto-generated method stub

try {

PrintWriter out = response.getWriter();

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

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

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

session.close();

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

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

} catch (Exception ex) {

throw ex;

}

}

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Config.hbm.xml:**

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateConfig</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>InitHibernate</servlet-name>

<servlet-class>InitHibernate</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>InitHibernate</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

1. **Configure Hibernate using XML in Eclipse IDE.**

Source Code:

<?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.jdbc.Driver</property>

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

<property name="connection.username">mysql user id</property>

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

</session-factory>

</hibernate-configuration>

Code:

package com.ecommerce;

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;

}

}

HTML:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate With XML</title>

</head>

<body>

<a href="list”>List Products</a><br>

</body>

</html>

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Date;

public class EProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

public EProduct() {

}

public EProduct(long id, String name, BigDecimal price, Date dateAdded) {

this.ID = id;

this.name = name;

this.price = price;

this.dateAdded = dateAdded;

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

}

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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 javax.transaction.\*;

import javax.xml.bind.\*;

import java.io.Serializable;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.ecommerce.EProduct;

import com.ecommerce.HibernateUtil;

/\*\*

\* Servlet implementation class ListProducts

\*/

@WebServlet("/ListProducts")

public class ListProducts extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ListProducts() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

try {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.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;

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package="com.ecommerce">

<class name="EProduct" table="eproduct">

<id name="ID" column="ID">

<generator class="increment"/>

</id>

<property name="name" type="string" column="NAME"/>

<property name="price" type="big\_decimal" column="PRICE"/>

<property name="dateAdded" type="timestamp" column="DATE\_ADDED"/>

</class>

</hibernate-mapping>

Config.hbm.xml:

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateXML</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ListProducts</servlet-name>

<servlet-class>ListProducts</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ListProducts</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

**3)Configure Hibernate using Annotations in Eclipse IDE**.

**SOURCE CODE:**

package com.ecommerce;

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;

}

}

**HTML CODE:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Annotation Example</title>

</head>

<body>

<a href="list">List Products</a><br>

</body>

</html>

**Annotations Code:**

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name= "eproduct")

public class EProduct {

@Id @GeneratedValue

@Column(name = "ID")

private long ID;

@Column(name = "name")

private String name;

@Column(name = "price")

private BigDecimal price;

@Column(name = "date\_added")

private Date dateAdded;

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

}

**Servelet Code:**

mport java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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 javax.transaction.\*;

import javax.xml.bind.\*;

import java.io.Serializable;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.ecommerce.EProduct;

import com.ecommerce.HibernateUtil;

/\*\*

\* Servlet implementation class ListProducts

\*/

@WebServlet("/ListProducts")

public class ListProducts extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ListProducts() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

try {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.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;

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

**XML CODE:**

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateAnnotations</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ListProducts</servlet-name>

<servlet-class>ListProducts</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ListProducts</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

**4) Demonstrate Hibernate logging by Log4j.**

<?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.jdbc.Driver</property>

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

<property name="connection.username">mysql user id</property>

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

</session-factory>

</hibernate-configuration>

**log4j.appender**

# Direct log messages to a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=/home/oem/Downloads/hibernate.log

log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

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

log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Direct log messages to stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

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

log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Root logger option

log4j.rootLogger=INFO, file, stdout

# Log everything. Good for troubleshooting

log4j.logger.org.hibernate=INFO

# Log all JDBC parameters

log4j.logger.org.hibernate.type=ALL

# Direct log messages to a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=/home/oem/Downloads/hibernate.log

log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

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

log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Direct log messages to stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

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

log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Root logger option

log4j.rootLogger=INFO, file, stdout

# Log everything. Good for troubleshooting

log4j.logger.org.hibernate=INFO

# Log all JDBC parameters

log4j.logger.org.hibernate.type=ALL

HTML CODE:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Configuration Example</title>

</head>

<body>

<a href="init">Initialize Hibernate</a><br>

</body>

</html>

HiberNate Code:

package com.ecommerce;

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;

}

}

**SERVELET CODE:**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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 javax.transaction.\*;

import javax.xml.bind.\*;

import java.io.Serializable;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.ecommerce.EProduct;

import com.ecommerce.HibernateUtil;

/\*\*

\* Servlet implementation class InitHibernate

\*/

@WebServlet("/InitHibernate")

public class InitHibernate extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public InitHibernate() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

try {

PrintWriter out = response.getWriter();

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

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

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

session.close();

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

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

} catch (Exception ex) {

throw ex;

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

XML code:

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateLogging</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>InitHibernate</servlet-name>

<servlet-class>InitHibernate</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>InitHibernate</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

**5) Demonstrate mapping List, Set, Bag, and Map in collection using XML file.**

DATABASE CODE:

•

• DROP TABLE IF EXISTS `colors`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `colors` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `color\_name` varchar(40) DEFAULT NULL,

• `idx` int(11) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `colors`

• --

•

• LOCK TABLES `colors` WRITE;

• /\*!40000 ALTER TABLE `colors` DISABLE KEYS \*/;

• INSERT INTO `colors` VALUES (1,'Red',0,1),(2,'Silver',1,1),(3,'Gray',0,2),(4,'White',1,2),(5,'Maroon',0,3);

• /\*!40000 ALTER TABLE `colors` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `eproduct`

• --

•

• DROP TABLE IF EXISTS `eproduct`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `eproduct` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `name` varchar(100) DEFAULT NULL,

• `price` decimal(10,2) DEFAULT NULL,

• `date\_added` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `eproduct`

• --

•

• LOCK TABLES `eproduct` WRITE;

• /\*!40000 ALTER TABLE `eproduct` DISABLE KEYS \*/;

• INSERT INTO `eproduct` VALUES (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19');

• /\*!40000 ALTER TABLE `eproduct` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `finance`

• --

•

• DROP TABLE IF EXISTS `finance`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `finance` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `ftype` varchar(10) DEFAULT NULL,

• `name` varchar(30) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `finance`

• --

•

• LOCK TABLES `finance` WRITE;

• /\*!40000 ALTER TABLE `finance` DISABLE KEYS \*/;

• INSERT INTO `finance` VALUES (1,'CREDITCARD','EMI on Citibank Card',1),(3,'BANK','40% finance from SBI',2),(4,'BANK','60% finance from ICICI',3),(5,'BANK','20% finance from ICICI',1);

• /\*!40000 ALTER TABLE `finance` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `os`

• --

•

• DROP TABLE IF EXISTS `os`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `os` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `name` varchar(30) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `os`

• --

•

• LOCK TABLES `os` WRITE;

• /\*!40000 ALTER TABLE `os` DISABLE KEYS \*/;

• INSERT INTO `os` VALUES (1,'Windows 10',1),(2,'Windows 10',2),(3,'FreeDOS',2),(4,'RedHat Linux',2),(5,'Windows 10',3);

• /\*!40000 ALTER TABLE `os` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `screensizes`

• --

•

• DROP TABLE IF EXISTS `screensizes`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `screensizes` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `size` varchar(10) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `screensizes`

• --

•

• LOCK TABLES `screensizes` WRITE;

• /\*!40000 ALTER TABLE `screensizes` DISABLE KEYS \*/;

• INSERT INTO `screensizes` VALUES (1,'12 in',1),(2,'14.5 in',2),(3,'14.9 in',2),(4,'15.5 in',3);

• /\*!40000 ALTER TABLE `screensizes` ENABLE KEYS \*/;

• UNLOCK TABLES;

HIBERNATE CODE:

package com.ecommerce;

public class Color {

private long COLORID;

private String name;

public Color() {

}

public Color(String name) {

this.COLORID = 0;

this.name = name;

}

public long getCOLORID() {return this.COLORID; }

public String getName() { return this.name;}

public void setCOLORID(long id) { this.COLORID = id;}

public void setName(String name) { this.name = name;}

}

**HIBERNATE:**

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class EProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

private List<Color> colors;

private Collection<ScreenSizes> screenSizes;

private Set<OS> os;

private Map finance;

public EProduct() {

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public List<Color> getColors() { return this.colors;}

public Collection<ScreenSizes> getScreensizes() { return this.screenSizes;}

public Set<OS> getOs() { return this.os;}

public Map getFinance() { return this.finance;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

public void setColors(List<Color> colors) { this.colors = colors;}

public void setScreensizes(Collection<ScreenSizes> sizes) { this.screenSizes = sizes;}

public void setOs(Set<OS> os) { this.os = os;}

public void setFinance(Map finance) { this.finance = finance;}

}

XML CODE:

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package="com.ecommerce">

<class name="Color" table="colors">

<id name="COLORID" type="long" column="ID">

<generator class="identity"/>

</id>

<property name="name" type="string" column="COLOR\_NAME"/>

</class>

</hibernate-mapping>

**6) Demonstrate lazy collection in Hibernate.**

**DATABASE CODE:**

•

• DROP TABLE IF EXISTS `colors`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `colors` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `color\_name` varchar(40) DEFAULT NULL,

• `idx` int(11) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `colors`

• --

•

• LOCK TABLES `colors` WRITE;

• /\*!40000 ALTER TABLE `colors` DISABLE KEYS \*/;

• INSERT INTO `colors` VALUES (1,'Red',0,1),(2,'Silver',1,1),(3,'Gray',0,2),(4,'White',1,2),(5,'Maroon',0,3);

• /\*!40000 ALTER TABLE `colors` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `eproduct`

• --

•

• DROP TABLE IF EXISTS `eproduct`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `eproduct` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `name` varchar(100) DEFAULT NULL,

• `price` decimal(10,2) DEFAULT NULL,

• `date\_added` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `eproduct`

• --

•

• LOCK TABLES `eproduct` WRITE;

• /\*!40000 ALTER TABLE `eproduct` DISABLE KEYS \*/;

• INSERT INTO `eproduct` VALUES (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19');

• /\*!40000 ALTER TABLE `eproduct` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `finance`

• --

•

• DROP TABLE IF EXISTS `finance`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `finance` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `ftype` varchar(10) DEFAULT NULL,

• `name` varchar(30) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `finance`

• --

•

• LOCK TABLES `finance` WRITE;

• /\*!40000 ALTER TABLE `finance` DISABLE KEYS \*/;

• INSERT INTO `finance` VALUES (1,'CREDITCARD','EMI on Citibank Card',1),(3,'BANK','40% finance from SBI',2),(4,'BANK','60% finance from ICICI',3),(5,'BANK','20% finance from ICICI',1);

• /\*!40000 ALTER TABLE `finance` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `os`

• --

•

• DROP TABLE IF EXISTS `os`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `os` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `name` varchar(30) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `os`

• --

•

• LOCK TABLES `os` WRITE;

• /\*!40000 ALTER TABLE `os` DISABLE KEYS \*/;

• INSERT INTO `os` VALUES (1,'Windows 10',1),(2,'Windows 10',2),(3,'FreeDOS',2),(4,'RedHat Linux',2),(5,'Windows 10',3);

• /\*!40000 ALTER TABLE `os` ENABLE KEYS \*/;

• UNLOCK TABLES;

•

• --

• -- Table structure for table `screensizes`

• --

•

• DROP TABLE IF EXISTS `screensizes`;

• /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

• /\*!40101 SET character\_set\_client = utf8 \*/;

• CREATE TABLE `screensizes` (

• `ID` bigint(20) NOT NULL AUTO\_INCREMENT,

• `size` varchar(10) DEFAULT NULL,

• `product\_id` bigint(20) DEFAULT NULL,

• PRIMARY KEY (`ID`)

• ) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

• /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

•

• --

• -- Dumping data for table `screensizes`

• --

•

• LOCK TABLES `screensizes` WRITE;

• /\*!40000 ALTER TABLE `screensizes` DISABLE KEYS \*/;

• INSERT INTO `screensizes` VALUES (1,'12 in',1),(2,'14.5 in',2),(3,'14.9 in',2),(4,'15.5 in',3);

• /\*!40000 ALTER TABLE `screensizes` ENABLE KEYS \*/;

• UNLOCK TABLES;

**HIBERNATE CODE:**

package com.ecommerce;

public class Color {

private long COLORID;

private String name;

public Color() {

}

public Color(String name) {

this.COLORID = 0;

this.name = name;

}

public long getCOLORID() {return this.COLORID; }

public String getName() { return this.name;}

public void setCOLORID(long id) { this.COLORID = id;}

public void setName(String name) { this.name = name;}

**HIBERNATE CODE:**

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class EProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

private List<Color> colors;

private Set<Finance> finance;

private PDescription pdescrip;

public EProduct() {

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public List<Color> getColors() { return this.colors;}

public Set<Finance> getFinance() { return this.finance;}

public PDescription getPdescrip() { return this.pdescrip;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

public void setColors(List<Color> colors) { this.colors = colors;}

public void setFinance(Set<Finance> finance) { this.finance = finance;}

public void setPdescrip(PDescription pdescrip) { this.pdescrip = pdescrip;}

}

**HIBERNATE CODE:**

package com.ecommerce;

public class Finance {

private long FINANCEID;

private String name;

private String ftype;

public Finance() {

}

public Finance(String name, String ftype) {

this.FINANCEID = 0;

this.name = name;

this.ftype = ftype;

}

public long getFINANCEID() {return this.FINANCEID; }

public String getName() { return this.name;}

public String getFtype() { return this.ftype;}

public void setFINANCEID(long id) { this.FINANCEID = id;}

public void setName(String name) { this.name = name;}

public void setFtype(String ftype) { this.ftype= ftype;}

}

**HIBERNATE LAZY:**

package com.ecommerce;

public class OS {

private long OSID;

private String name;

public OS() {

}

public OS(String name) {

this.OSID = 0;

this.name = name;

}

public long getOSID() {return this.OSID; }

public String getName() { return this.name;}

public void setOSID(long id) { this.OSID = id;}

public void setName(String name) { this.name= name;}

}

**XML CODE:**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package="com.ecommerce">

<class name="Color" table="colors">

<id name="COLORID" type="long" column="ID">

<generator class="identity"/>

</id>

<property name="name" type="string" column="COLOR\_NAME"/>

</class>

</hibernate-mapping>

**XML CODE:**

code:

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package="com.ecommerce">

<class name="EProduct" table="eproduct">

<id name="ID" type="long" column="ID">

<generator class="identity"/>

</id>

<property name="name" type="string" column="NAME"/>

<property name="price" type="big\_decimal" column="PRICE"/>

<property name="dateAdded" type="timestamp" column="DATE\_ADDED"/>

<list name="colors" cascade="all" lazy="true">

<key column="product\_id" />

<list-index column="idx" />

<one-to-many class="com.ecommerce.Color" />

</list>

<bag name="screensizes" cascade="all" lazy="true">

<key column="product\_id"></key>

<one-to-many class="com.ecommerce.ScreenSizes"/>

</bag>

<set name = "os" cascade="all" lazy=”true”>

<key column = "product\_id"/>

<one-to-many class="OS"/>

</set>

<map name = "finance" cascade="all">

<key column = "product\_id"/>

<index column = "ftype" type = "string"/>

<one-to-many class="com.ecommerce.Finance"/>

</map>

</class>

</hibernate-mapping>

**7) Demonstrate component mapping in Hibernate.**

CREATE TABLE `eproduct` (

`ID` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` varchar(100) DEFAULT NULL,

`price` decimal(10,2) DEFAULT NULL,

`date\_added` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`parts\_hdd` varchar(10) DEFAULT NULL,

`parts\_cpu` varchar(10) DEFAULT NULL,

`parts\_ram` varchar(10) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `eproduct`

--

**Code:**

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class EProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

private ProductParts parts;

public EProduct() {

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public ProductParts getParts() { return this.parts;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

public void setParts(ProductParts parts) { this.parts = parts;}

}

package com.ecommerce;

public class ProductParts {

private String hdd;

private String cpu;

private String ram;

public String getHdd() { return this.hdd;}

public String getCpu() { return this.cpu;}

public String getRam() { return this.ram;}

public void setHdd(String value) { this.hdd= value;}

public void setCpu(String value) { this.cpu= value;}

public void setRam(String value) { this.ram= value;}

}

package com.ecommerce;

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;

}

}

**HTML:**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package="com.ecommerce">

<class name="EProduct" table="eproduct">

<id name="ID" type="long" column="ID">

<generator class="identity"/>

</id>

<property name="name" type="string" column="NAME"/>

<property name="price" type="big\_decimal" column="PRICE"/>

<property name="dateAdded" type="timestamp" column="DATE\_ADDED"/>

<component name="parts" class="com.ecommerce.ProductParts">

<property name="hdd" column="parts\_hdd" type="string" />

<property name="cpu" column="parts\_cpu" type="string" />

<property name="ram" column="parts\_ram" type="string" />

</component>

</class>

</hibernate-mapping>

<?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.jdbc.Driver</property>

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

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

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

<mapping resource="com/ecommerce/EProduct.hbm.xml"/>

</session-factory>

</hibernate-configuration>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Component Mapping</title>

</head>

<body>

<a href="details">Product Details</a><br>

</body>

</html>

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

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 javax.transaction.\*;

import javax.xml.bind.\*;

import java.io.Serializable;

import java.math.BigDecimal;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Collection;

import java.util.List;

import java.util.Map;

import java.util.Set;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.ecommerce.EProduct;

import com.ecommerce.HibernateUtil;

import com.ecommerce.ProductParts;

/\*\*

\* Servlet implementation class ProductDetails

\*/

@WebServlet("/ProductDetails")

public class ProductDetails extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ProductDetails() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

try {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

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

PrintWriter out = response.getWriter();

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

out.println("<b>Component Mapping</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());

ProductParts parts = p.getParts();

out.println("Parts =" + parts.getCpu() + ", " + parts.getHdd() + ", " + parts.getRam());

out.println("<hr>");

}

session.close();

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

} catch (Exception ex) {

throw ex;

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

// TODO Auto-generated method stub

doGet(request, response);

}

}

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateComponentMapping</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ProductDetails</servlet-name>

<servlet-class>ProductDetails</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProductDetails</servlet-name>

<url-pattern>/details</url-pattern>

</servlet-mapping>

</web-app>

**8)** **Demonstrate integration of Hibernate with spring.**

**HTML:**

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>SpringHibernateWeb</groupId>

<artifactId>SpringHibernateWeb</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpringHibernateWeb Maven Webapp</name>

<url>http://maven.apache.org</url>

<!-- JBoss repository for Hibernate -->

<repositories>

<repository>

<id>JBoss repository</id>

<url>http://repository.jboss.org/nexus/content/groups/public/</url>

</repository>

</repositories>

<properties>

<org.springframework.version>3.0.5.RELEASE</org.springframework.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hibernate.javax.persistence</groupId>

<artifactId>hibernate-jpa-2.1-api</artifactId>

<version>1.0.0.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.16</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>3.6.3.Final</version>

</dependency>

<dependency>

<groupId>javassist</groupId>

<artifactId>javassist</artifactId>

<version>3.12.1.GA</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.9</version>

</dependency>

</dependencies>

<build>

<finalName>SpringHibernateWeb</finalName>

</build>

</project>

DROP TABLE IF EXISTS `eproduct`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `eproduct` (

`ID` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` varchar(100) DEFAULT NULL,

`price` decimal(10,2) DEFAULT NULL,

`date\_added` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`parts\_hdd` varchar(10) DEFAULT NULL,

`parts\_cpu` varchar(10) DEFAULT NULL,

`parts\_ram` varchar(10) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

INSERT INTO `eproduct` VALUES (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57','2 Gb HDD','AMD Phenom','4 Gb'),

(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07','500 Gb HDD','Core-i7','4 Gb'),

(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19','1 Tb HDD','Core-i7','8 Gb');

/\*!40000 ALTER TABLE `eproduct` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

**Code:**

package com.ecommerce.entity;

import java.math.BigDecimal;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name= "eproduct")

public class EProductEntity {

@Id @GeneratedValue

@Column(name = "ID")

private long ID;

@Column(name = "name")

private String name;

@Column(name = "price")

private BigDecimal price;

@Column(name = "date\_added")

private Date dateAdded;

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

}

Step 3.8.5: Creating a DAO class EProductDAO

package com.ecommerce.dao;

import java.util.List;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.ecommerce.entity.EProductEntity;

@Repository

public class EProductDAO {

@Autowired

private SessionFactory sessionFactory;

@SuppressWarnings("unchecked")

public List<EProductEntity> getAllProducts() {

return this.sessionFactory.getCurrentSession().createQuery("from EProducts").list();

}

}

package com.ecommerce.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import com.ecommerce.entity.EProductEntity;

import com.ecommerce.service.EProductService;

@Controller

public class EProductController {

@Autowired

private EProductService eproductService;

@RequestMapping(value = "/productList", method = RequestMethod.GET)

public String listProducts(ModelMap map)

{

map.addAttribute("eproduct", new EProductEntity());

map.addAttribute("productList", eproductService.getAllProducts());

return "productList";

}

}

**Web.xml:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:jee="http://www.springframework.org/schema/jee"

xmlns:lang="http://www.springframework.org/schema/lang"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop/ http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context/ http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/jee/ http://www.springframework.org/schema/jee/spring-jee.xsd

http://www.springframework.org/schema/lang/ http://www.springframework.org/schema/lang/spring-lang.xsd

http://www.springframework.org/schema/tx/ http://www.springframework.org/schema/tx/spring-tx.xsd

http://www.springframework.org/schema/util/ http://www.springframework.org/schema/util/spring-util.xsd">

<context:annotation-config />

<context:component-scan base-package="com.ecommerce.controller" />

<bean id="jspViewResolver"

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass"

value="org.springframework.web.servlet.view.JstlView"></property>

<property name="prefix" value="/WEB-INF/view/"></property>

<property name="suffix" value=".jsp"></property>

</bean>

<bean id="messageSource"

class="org.springframework.context.support.ReloadableResourceBundleMessageSource">

<property name="basename" value="classpath:messages"></property>

<property name="defaultEncoding" value="UTF-8"></property>

</bean>

<bean id="propertyConfigurer"

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"

p:location="/WEB-INF/jdbc.properties"></bean>

<bean id="dataSource"

class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"

p:driverClassName="${jdbc.driverClassName}"

p:url="${jdbc.databaseurl}" p:username="${jdbc.username}"

p:password="${jdbc.password}"></bean>

<bean id="sessionFactory"

class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

<property name="dataSource" ref="dataSource"></property>

<property name="configLocation">

<value>classpath:hibernate.cfg.xml</value>

</property>

<property name="configurationClass">

<value>org.hibernate.cfg.AnnotationConfiguration</value>

</property>

<property name="hibernateProperties">

<props>

<prop key="hibernate.dialect">${jdbc.dialect}</prop>

<prop key="hibernate.show\_sql">true</prop>

</props>

</property>

</bean>

<bean id="eproductDAO" class="com.ecommerce.dao.EProductDAO"></bean>

<bean id="eproductService" class="com.ecommerce.service.EProductService"></bean>

<tx:annotation-driven />

<bean id="transactionManager"

class="org.springframework.orm.hibernate3.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

</beans>

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.dialect=org.hibernate.dialect.MySQLDialect

jdbc.databaseurl=jdbc:mysql://127.0.0.1:3306/ecommerce

jdbc.username=userid

jdbc.password=password

<%@taglib uri="http://www.springframework.org/tags" prefix="spring"%>

<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<html>

<head>

<title>Spring With Hibernate</title>

</head>

<body>

<h3>Product List </h3>

<c:if test="${!empty productList}">

<table class="data">

<tr>

<th>Name</th>

<th>Price</th>

<th>Date Added</th>

</tr>

<c:forEach items="${productList}" var="product">

<tr>

<td>${product.name} </td>

<td>${product.price}</td>

<td>${product.date\_added}</td>

</tr>

</c:forEach>

</table>

</c:if>

</body>

</html>

<html>

<body>

<h2>Spring With Hibernate</h2>

<a href="/productList">Product List</a>

</body>

</html>

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

<!DOCTYPE hibernate-configuration PUBLIC

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

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<mapping class="com.ecommerce.entity.EProductEntity"></mapping>

</session-factory>

</hibernate-configuration>

<!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>

<welcome-file-list>

<welcome-file>/WEB-INF/view/index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>eproduct</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>eproduct</servlet-name>

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

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/eproduct-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

</web-app>