

4.Demonstrate the hibernate by logging by log4j.

Configuring hibernate with 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.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>
```

Configuring log4j with log4j.properties:

```
# 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
```

- Be sure to change the path of the log file log4j.appender.file.File=xxxxx to an actual path in your disk

Creating an HTML page index.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>
```

Creating an hibernateUtil class to initiate hibernate in 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;
}
}

```

Creating an InitHibernate servlet:

```

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);
    }

}

```

Configuring web.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>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>

```

