Configure Hibernate using XML in Eclipse IDE.

Index.html

```
<html>
<head> </head>
<body>
<h3>Hibernate Configuration Example </h3>
<a href="init">Initialize Hibernate</a>
<br>
<br>
<br>
<h3>Hibernate Query Demo </h3> $0 ==
<a href="query-demo">Query Here and get the data</a> <br>
</body>
</html>
```

Web.xml:

```
<?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>
```

Com.ecommerce:

```
package practice;



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 ID;
    }

    public void setID(long iD) {
        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;
    }

    public Date getDateAdded() {
        return dateAdded;
    }

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

}
```

Com.simpl:

```
package practice;


package com.simpli;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import org.hibernate.*;

@WebServlet("/init")
public class Hibernateinit extends HttpServlet {

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

```
            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            // STEP 1: Get a Session (connection) from the Session Factory
class

            SessionFactory factory = HibernateUtil.getSessionFactory();

            // STE2 Create the session object
            Session session = factory.openSession();

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

            session.close();

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

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

}
```

Query:

```
package practice;


import java.io.*;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import org.hibernate.*;

import com.ecommerce.Eproduct;

@WebServlet("/query-demo")
public class HibernateQueryDemo extends HttpServlet {

      protected void doGet(HttpServletRequest request, HttpServletResponse
response)
                  throws ServletException, IOException {
            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            // STEP 1: Get a Session (connection) from the Session Factory
class

            SessionFactory factory = HibernateUtil.getSessionFactory();

            // STE2 Create the session object
            Session session = factory.openSession();

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

            // STEP 3 Query the DB and get the data
            List<Eproduct> eproducts = session.createQuery("from
Eproduct").list();
```

```
            out.println("<table border=1>");
            for (Eproduct prod : eproducts) {
                  out.println("<tr>" + "<td>" + prod.getID() + "<td>" +
prod.getName() + "<td>" + prod.getPrice() + "<td>"
                              + prod.getDateAdded());

            }
            out.println("</table>");

            session.close();

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

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

}
```

Util:

```
package practice;


package com.simpli;

import org.hibernate.SessionFactory;
import org.hibernate.boot.*;
import org.hibernate.boot.registry.*;

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;
      }

}
```