Assisted Project 7: Demonstrate component mapping in Hibernate.

Index:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hibernate Component Mapping</title>
</head>
<body>
<a href="ProductDetails">Product Details</a><br>

</body>
</html>
```

Hibernate.cfg.xml:

```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">root</property>
   <property name="connection.password">Vinny@12345</property>
   <mapping resource="com/ecommerce/EProduct.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

EProduct.hbm.xml:

```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" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
    <property name="price" type="big_decimal" column="PRICE"/>
```

```xml
                <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>
```

EProduct.java:

```java
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;}
}
```

ProductParts.java:

```java
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;}

}
```

HibernateUtil.java:

```java
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) {
```

```java
            throw new ExceptionInInitializerError(th);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

ProductDetails.java:

```java
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;
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
    }
```

```java
    /**
     * @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)
     */
```

```java
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}
```

File   Edit   Navigate   Search   Project   Run   Window   Help

Project Explorer

> Hibernate4
> Hibernate5
> Hibernate6
> Hibernate7
  > Deployment Descriptor: Hibernate7
  > JAX-WS Web Services
  > Java Resources
    > src/main/java
      > (default package)
      > com.ecommerce
        > EProduct.java
        > HibernateUtil.java
        > ProductParts.java
        EProduct.hbm.xml
      hibernate.cfg.xml
    > Libraries
  > build
  > src
    > main
      > java
      > webapp
        > META-INF
        > WEB-INF
          > lib
          index.html
> HibernateSample
> HTTPSessionDemo
> InnerClassDemo
> JDBC2
> JDBC4
> JDBC5
> JDBC6

ProductParts.ja   HibernateUtil.j   hibernate.cfg.x   index.html   ProductDetails.   http://localhos

http://localhost:8080/Hibernate7/ProductDetails

Component Mapping
ID: 1, Name: HP Laptop ABC, Price: 21900.00, Date Added: 2019-06-04 12:48:57.0 Parts =AMD Phenom, 2 Gb HDD, 4 Gb

ID: 2, Name: Acer Laptop ABC, Price: 23300.00, Date Added: 2019-06-04 12:49:07.0 Parts =Core-i7, 500 Gb HDD, 4 Gb

ID: 3, Name: Lenovo Laptop ABC, Price: 33322.00, Date Added: 2019-06-04 12:49:19.0 Parts =Core-i7, 1 Tb HDD, 8 Gb

Outline
There is no active editor
that provides an outline.

Markers   Properties   Servers   Data Source Explorer   Snippets   Terminal   Console

apache-tomcat-8.5.85 at localhost [Apache Tomcat] C:\Users\kvsivasankar\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-150

May 30, 2023 6:51:52 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
ANTLR Tool version 4.10.1 used for code generation does not match the current runtime version 4.12.0
ANTLR Runtime version 4.10.1 used for parser compilation does not match the current runtime version 4.12.0
ANTLR Tool version 4.10.1 used for code generation does not match the current runtime version 4.12.0
ANTLR Runtime version 4.10.1 used for parser   Superscript s not match the current runtime version 4.12.0

Done

Type here to search   30°C   ENG   18:58
                                                                        IN   30-05-2023