

## Demonstrate lazy collection in Hibernate.Code:

Ecommerce.com:

```
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;}
}
}
```

## E product.com:

```
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 Set<Finance>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 Set<Finance> 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(Set<Finance> finance) { this.finance = finance;}

}

```

### Finance.com:

```

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;}

}

```

### Hibernateutil:

```

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;
        }
    }
}

```

**Os.com:**

```

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;}
}

```

**Productdetails.com:**

```

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.Color;
import com.ecommerce.EProduct;
import com.ecommerce.Finance;
import com.ecommerce.HibernateUtil;
//import com.ecommerce.PDescription;

/**
 * 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
        //response.getWriter().append("Served at:
    ").append(request.getContextPath());
        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>One to One 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());
//            PDescription descrip = p.getPdescrip();
//            out.println("<br>Description:" +
descrip.getDescrip());
//            out.println("<hr>");
        }
        out.println("<b>One to Many and Many to One 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());
            List<Color> colors = p.getColors();
            out.println("<br>Colors: <ul>");
            for (Color c: colors) {
                out.print("<li>" + c.getName() + "</li>");
            }
            out.println("</ul>");
            out.println("<hr>");
        }
        out.println("<b>Many to Many 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());
            Set<Finance> finances= p.getFinance();
            out.println("<br>Finance Options : <ul>");
            for (Finance f: finances) {
                out.print("<li>" + f.getFtype() + "</li>");
            }

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

```

## Screensize:

```

package com.ecommerce;

public class ScreenSizes {
    private long SCREENID;
    private String size;

    public ScreenSizes() {

    }

    public ScreenSizes(String size) {
        this.SCREENID = 0;
        this.size = size;
    }

    public long getSCREENID() {return this.SCREENID; }
    public String getSize() { return this.size;}
    public void setSCREENID(long id) { this.SCREENID = id;}
    public void setSize(String size) { this.size = size;}
}

```

## Index.html:

```

<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Collection Mapping</title>

</head>

<body> <a href "details">details</a>

<br>

</body>

</html>

```

**Colors:**

```
<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>
```

**Eproduct:**

```
<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"/>
<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>
<set name="finance" cascade="all" lazy="true">
<key column="product_id"/>
<one-to-many class="com.ecommerce.Finance"/>
</set>
</class>
</hibernate-mapping>
```

**Finance:**

```
<hibernate-mapping package="com.ecommerce">
<class name="Finance" table="finance">
<id name="FINANCEID" type="long" column="ID">
<generator class="identity"/>
</id>
<property name="name" type="string" column="NAME"/>
<property name="ftype" type="string" column="FTYPE"/>
</class>
</hibernate-mapping>
```

**Driver:**

```
<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/assisted3phase2</property>
>
<property name="connection.username">root</property>
```

```

<property name="connection.password">27@Salma</property>
<property name="dialect">org.hibernate.dialect.MySQL57Dialect</property>
<mapping resource="com/ecommerce/EProduct.hbm.xml"/>
<mapping resource="com/ecommerce/Color.hbm.xml"/>
<mapping resource="com/ecommerce/ScreenSizes.hbm.xml"/>
<mapping resource="com/ecommerce/Os.hbm.xml"/>
<mapping resource="com/ecommerce/Finance.hbm.xml"/>
</session-factory>
</hibernate-configuration>

```

#### Os:

```

<hibernate-mapping package="com.ecommerce">
<class name="OS" table="os">
<id name="OSID" type="long" column="ID">
<generator class="identity"/>
</id>
<property name="name" type="string" column="NAME"/>
</class>
</hibernate-mapping>

```

#### Screen size:

```

<hibernate-mapping package="com.ecommerce">
<class name="ScreenSizes" table="screensizes">
<id name="SCREENID" type="long" column="ID">
<generator class="identity"/>
</id>
<property name="size" type="string" column="SIZE"/>
</class>
</hibernate-mapping>

```

#### Web.xml:

```

<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>ph2-As-set3-prj6</display-name>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.jsp</welcome-file>
<welcome-file>default.htm</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>

```