Component Mapping

// Creating a table eproduct in the database and filling it with sample data

```
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES,
SQL_NOTES=0 */;

--
-- 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,
  `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`
--

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 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_RESUL
TS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
```

```
*/;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2019-06-07 10:05:19
```

```
//Creating a class EProduct

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;}
}
```

```
//Creating a class ProductParts
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;}

}
```

```
//Creating a 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 a hibernate table configuration file EProduct.hbm.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"/>
    <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>
```

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

//Creating an HTML page index.html

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

</body>
</html>
```

//Creating a ProductDetails 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.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);
    }

}
```

```
//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>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>
```