

## SOURCE CODE FOR ADDING A PRODUCT TO DATABASE USING HYBERNATE

//////////////////////////////////// index page

```
<!DOCTYPE html>
<html>
  <body>
    <center>
      <legend>Product details</legend>
      <form method="get" action="productDao">
        Enter Product name : <input type="text" name="pname" required><br>
        Enter price : <input type="text" name="price" required ><br>
        <input type="submit" value="Submit">
      </form>
    </center>
  </body>
</html>
```

////////// ProductDAO code

```
package Shop;

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

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class productDao extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html");        // create a configuration object
        Configuration config = new Configuration().configure();

        // creating a session factory object
        SessionFactory sf = config.buildSessionFactory();

        // create a session object and opened it
        Session session = sf.openSession();

        //begin a transaction
        session.beginTransaction();

        //Read two values from html form since id is automatically assigned by sequel.
        String pname = request.getParameter("pname");
        float pprice = Float.parseFloat(request.getParameter("price"));

        //put all values in Employee Object
        product p =new product();

        //add to prdoducts
        p.setPname(pname);
        p.setPrice(pprice);

        // add them to session object
        int i = (int) session.save(p);

        // session commit
        session.getTransaction().commit();
        PrintWriter out = response.getWriter();
        if(i>0) {
            out.println("Record inserted<br><a href='\"index.html\"'>back</a>");
        }
        else
            out.println("Record not inserted<br><a href='\"index.html\"'>back</a>");

        session.close();

    }
}
```

//////////////////////////////////// Product persist class

```
package Shop;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity

@Table
public class product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int pid;
    @Column(length = 20)
    private String pname;
    @Column(length = 7)
    private float price;

    public int getPid() {
        return pid;
    }
    public void setPid(int pid) {
        this.pid = pid;
    }
    public String getPname() {
        return pname;
    }
    public void setPname(String pname) {
        this.pname = pname;
    }
    public float getPrice() {
        return price;
    }
    public void setPrice(float price) {
        this.price = price;
    }
}
```

//////////////////////////////////// hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!--
~ Hibernate, Relational Persistence for Idiomatic Java
~
~ License: GNU Lesser General Public License (LGPL), version 2.1 or later.
~ See the lgpl.txt file in the root directory or <http://www.gnu.org/licenses/lgpl-2.1.html>.
-->
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>

<property name="format_sql">false</property>

<!-- DB connection details should be provided here -->
<property name="connection.url">jdbc:mysql://localhost:3306/products</property>
<property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
<property name="connection.username">root</property>
<property name="connection.password">root</property>

<!-- JDBC connection pool (use the built-in) -->
<property name="connection.pool_size">4</property>

<!-- SQL dialect -->
<property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>

<!-- Disable the second-level cache -->
<property name="cache.provider_class">org.hibernate.cache.internal.NoCacheProvider</property>

<!-- Echo all executed SQL to stdout -->
<property name="show_sql">true</property>

<!-- Drop and re-create the database schema on startup -->
<property name="hbm2ddl.auto">update</property>
<!-- <property name="hbm2ddl.auto">update</property> -->

<mapping class="Shop.product"></mapping>

</session-factory>
</hibernate-configuration>
```

