# Project Name: Retrieving the Product Details Using the Product ID.

## GitHub Link:

<https://github.com/tamasjit/myjavaproject/tree/master/Phase%202/Phase-2_Practice_Project/q2_ProductApp>

## Source Code:

### index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>JDBC Product Details Project</title>

</head>

<body style="text-align: center">

<h1>Hello, Please enter product Id to continue.</h1>

<br/>

<form action="products" method="post">

    Product ID <br/><input type="text" name="productId"><br/>

    <input type="submit" value="Get Product Details">

</form>

</body>

</html>

### no.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Error</title>

</head>

<body style="text-align: center">

    <h1 style="text-align: center">No product found.</h1>

</body>

</html>

### ProductDetails.jsp

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>

<head>

    <style>

        .center {

            margin-left: auto;

            margin-right: auto;

        }

    </style>

    <title>Product Details</title>

</head>

<body style="text-align: center">

<table class="center">

    <tr>

        <th>Id</th>

        <th>Name</th>

        <th>Price</th>

    </tr>

    <c:forEach items="${requestScope.products}" var="prod">

        <tr>

            <td>${prod.getId()}</td>

            <td>${prod.getName()}</td>

            <td>${prod.getPrice()}</td>

        </tr>

    </c:forEach>

</table>

<br/>

</body>

</html>

### jdbc.properties:

url=jdbc:mysql://localhost:3306/ecommerce

user=root

password=Abc@123

### DbConnection.java:

package productdetails.jdbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DbConnection {

    private Connection dbConnection;

    public DbConnection(String url,String username, String password) throws ClassNotFoundException, SQLException {

        Class.forName("com.mysql.cj.jdbc.Driver");

        this.dbConnection = DriverManager.getConnection(url, username, password);

    }

    public Connection getConnection(){

        return this.dbConnection;

    }

    public void closeConnection() throws SQLException {

        if (this.dbConnection != null)

            this.dbConnection.close();

    }

}

### ProductsServlet.java

package jdbc\_productdetails;

import javax.servlet.\*;

import javax.servlet.http.\*;

import productdetails.jdbc.DbConnection;

import javax.servlet.annotation.\*;

import java.io.IOException;

import java.io.InputStream;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

@WebServlet(name = "ProductsServlet", value = "/products")

public class ProductsServlet extends HttpServlet {

    Properties props;

    @Override

    public void init() throws ServletException {

        super.init();

        InputStream dbDetails = this.getServletContext().getResourceAsStream("WEB-INF/jdbc.properties");

        props = new Properties();

        try {

            props.load(dbDetails);

            DbConnection connection = new DbConnection(props.getProperty("url"),props.getProperty("user"),props.getProperty("password"));

            Statement statement = connection.getConnection().createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

            statement.executeUpdate("insert into eproduct (name, price, date\_added) values ('Product1', 1000, now())");

            statement.executeUpdate("insert into eproduct (name, price, date\_added) values ('Product2', 2000, now())");

            statement.executeUpdate("insert into eproduct (name, price, date\_added) values ('Product3', 3000, now())");

            statement.executeUpdate("insert into eproduct (name, price, date\_added) values ('Product4', 4000, now())");

            statement.executeUpdate("insert into eproduct (name, price, date\_added) values ('Product5', 5000, now())");

            statement.close();

            connection.closeConnection();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    @Override

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

        String id = request.getParameter("productId");

        try {

            DbConnection connection = new DbConnection(props.getProperty("url"),props.getProperty("user"),props.getProperty("password"));

            Statement statement = connection.getConnection().createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

            ResultSet result = statement.executeQuery("select \* from eproduct where id =" + id);

            if (result.next() == false){

                response.sendRedirect("no.html");

            }else{

                result.beforeFirst();

                List<product> products = new ArrayList<>();

                while (result.next()){

                    product p = new product();

                    p.setId(Integer.parseInt(result.getString("id")));

                    p.setName(result.getString("name"));

                    p.setPrice(result.getFloat("price"));

                    products.add(p);

                }

                statement.close();

                connection.closeConnection();

                request.setAttribute("products",products);

                RequestDispatcher dispatcher = request.getRequestDispatcher("productDetails.jsp");

                dispatcher.forward(request,response);

            }

        } catch (SQLException | ClassNotFoundException e) {

            e.printStackTrace();

        }

    }

}

### Product.java

package jdbc\_productdetails;

public class product {

    private int id;

    private String name;

    private float price;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public float getPrice() {

        return price;

    }

    public void setPrice(float price) {

        this.price = price;

    }

}