

JSP MVC jdbc

Bước 0: Maven Project --> Add Dependencies

- Jsp
- Servlet: servlet – api
- Jdbc: Maria/Sql

<https://mvnrepository.com/artifact/jakarta.servlet/jakarta.servlet-api>

<https://mvnrepository.com/artifact/jakarta.servlet.jsp/jakarta.servlet.jsp-api>

<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core>

<https://mvnrepository.com/artifact/jakarta.servlet.jsp.jstl/jakarta.servlet.jsp.jstl-api>

<https://mvnrepository.com/artifact/org.glassfish.web/jakarta.servlet.jsp.jstl>

<https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client>

<https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc>

Bước 1: Tạo Entity: Product, ItemCart (package)

Product:

```
public class Product {  
    private int id;  
    private String name;  
    private double price;  
    private String image;
```

ItemCart

```
public class ItemCart {  
    private Product product;  
    private int quantity;
```

Bước 2: Tạo DAOImpl (Kết nối DB) (package)

- Resource (Tạo chuỗi kết nối): DataSource

Context.xml (webapp\META-INF)

https://tomcat.apache.org/tomcat-10.1-doc/jndi-datasource-examples-howto.html#Context_versus_GlobalNamingResources

- Tạo Class java: DAOImpl (Thực hiện các phương thức (query DB))

```
public class ProductDAOImp implements ProductDAO {  
    private DataSource datasource;  
    public ProductDAOImp(DataSource datasource) {  
        this.datasource = datasource;  
    }  
    @Override  
    public List<Product> findAll() {  
        String sql = "SELECT * FROM product";  
        List<Product> list = new ArrayList<Product>();  
        try {  
            Connection con = this.datasource.getConnection();  
            PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);  
            ResultSet rs = ps.executeQuery();
```

```

    } {
        while (rs.next()) {
            int id = rs.getInt("id");
            String name = rs.getString("name");
            Double price = rs.getDouble("price");
            String image = rs.getString("image");
            list.add(new Product(id, name, price, image));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return list;
}

@Override
public Product getById(int id) {

    String sql = "SELECT * FROM product WHERE id=?";
    Product p = null;
    try (
        Connection con = this.datasource.getConnection();
        PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);
    ){

        ps.setInt(1, id);

        try (ResultSet rs = ps.executeQuery();) {
            while (rs.next()) {
                String name = rs.getString("name");
                Double price = rs.getDouble("price");
                String image = rs.getString("image");
                p = new Product(id, name, price, image);
            }
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return p;
}

```

```

@Override
public void addProduct(Product p) {
    String sql = "INSERT INTO product (name, price, image) VALUES (?, ?, ?)";
    try (
        Connection con = this.datasource.getConnection();
        PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);
    ){

        ps.setString(1, p.getName());
        ps.setDouble(2, p.getPrice());
        ps.setString(3, p.getImage());
        ps.executeUpdate();

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

Bước 3: DAO gọi các phương thức DAOImpl (package)

Tạo Interface DAO

```

public interface ProductDAO {
    public List<Product> findAll(); // ko truc tiep DB
    public Product getById(int id);
    public void addProduct(Product p);
}

```

Bước 4: Controller (Servlet) (package)

ProductController.java:

```

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

    @Resource(name = "jdbc/shopping")
    private DataSource dataSource;

    private ProductDAO productDAO;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ProductController() {
        super();
        // TODO Auto-generated constructor stub
    }

    @Override
    public void init(ServletConfig config) throws ServletException {
        // TODO Auto-generated method stub
        super.init(config);
        try {
            System.out.println(this.dataSource.getConnection());
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        productDAO = new ProductDAOImp(this.dataSource);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        Product p1 = new Product("Product 1", 50000, "image1.png");
        productDAO.addProduct(p1);
        Product p2 = new Product("Product 2", 20000, "image2.png");
        productDAO.addProduct(p2);
        Product p3 = new Product("Product 3", 40000, "image3.jpeg");
        productDAO.addProduct(p3);
        Product p4 = new Product("Product 4", 80000, "image4.jpeg");
        productDAO.addProduct(p4);
        Product p5 = new Product("Product 5", 100000, "image5.png");
        productDAO.addProduct(p5);

        request.setAttribute("products", productDAO.findAll());

        request.getRequestDispatcher("views/product/index.jsp").forward(request, response);
    }
}

```

CartItemController.java

```

@WebServlet(urlPatterns = {"/cart", "/cart*"})
public class CartController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Resource(name = "jdbc/shopping")
    private DataSource dataSource;

    private ProductDAO productDAO;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CartController() {
        super();
        // TODO Auto-generated constructor stub
        System.out.println(dataSource);
    }

    @Override
    public void init(ServletConfig config) throws ServletException {
        // TODO Auto-generated method stub
        super.init(config);
        productDAO = new ProductDAOImp(this.dataSource);
    }
}

```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String action = request.getParameter("action") != null ? request.getParameter("action") : "";

    switch(action) {
        case "buy":
            doGetBuy(request, response);
            break;
        case "remove":
            doGetRemove(request, response);
            break;
        default:
            doGetDisplayCart(request, response);
            break;
    }
}

private void doGetDisplayCart(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    request.getRequestDispatcher("views/cart/index.jsp").forward(request, response);
}

private void doGetRemove(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
    List<ItemCart> cart = (List<ItemCart>) session.getAttribute("cart");
    int index = isProductExisting(Integer.parseInt(request.getParameter("id")), cart);
    cart.remove(index);
    session.setAttribute("cart", cart);
    response.sendRedirect("cart");
}

private void doGetBuy(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
    List<ItemCart> cart = null;

    if (session.getAttribute("cart") == null) {
        cart = new ArrayList<ItemCart>();
    } else {
        cart = (List<ItemCart>) session.getAttribute("cart");
    }
    int index = isProductExisting(Integer.parseInt(request.getParameter("id")), cart);
    if (index == -1) {
        cart.add(new ItemCart(productDAO.getById(Integer.parseInt(request.getParameter("id"))), 1));
    } else {
        int quantity = cart.get(index).getQuantity() + 1;
        cart.get(index).setQuantity(quantity);
    }
    session.setAttribute("cart", cart);

    response.sendRedirect("cart");
}

private int isProductExisting(int id, List<ItemCart> cart) {
    for (int i = 0; i < cart.size(); i++) {
        if (cart.get(i).getProduct().getId() == id) {
            return i;
        }
    }
    return -1;
}
}

```

Bước 5: Tạo các view (.jsp)

View Product (webapp\views\product\index.jsp)

```

<c:forEach var="product" items="${products}">
    <tr>
        <td>${product.id }</td>
        <td>${product.name }</td>
        <td>
            
        </td>
        <td>${product.price }</td>
        <td align="center">
            <a href="${pageContext.request.contextPath}/cart?action=buy&id=${product.id}">Buy</a>
        </td>
    </tr>
</c:forEach>

```

View Cart (webapp\views\cart\index.jsp)

```

<c:set var="total" value="0"></c:set>
<c:forEach var="item" items="${sessionScope.cart}">
  <c:set var="total"
    value="${total + item.product.price * item.quantity }"></c:set>
  <tr>
    <td align="center">
      <a href="${pageContext.request.contextPath }/cart?action=remove&id=${item.product.id}"
        onclick="return confirm('Are you sure?')">Remove</a></td>
    <td>${item.product.id }</td>
    <td>${item.product.name }</td>
    <td>
      
    </td>
    <td>${item.product.price }</td>
    <td>${item.quantity }</td>
    <td>${item.product.price * item.quantity }</td>
  </tr>
</c:forEach>

```

()