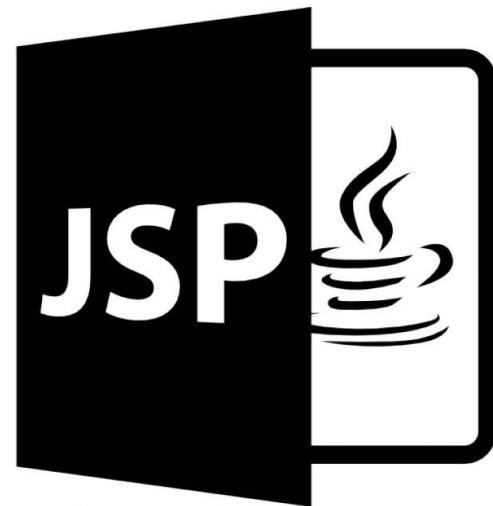


6장. MVC 웹 애플리케이션

▶▶ 상품 관리



상품 관리 프로젝트

➤ 상품 관리 프로젝트 만들기 – MVC 패턴

전체 상품 목록을 보여주고 상품을 선택하면 세부 정보를 보여주도록 구현함

1. Model(모델 구현) – Product.java, ProductService.java
2. View(뷰 구현) – productList.jsp, productInfo.jsp
3. Controller(컨트롤러 구현) – ProductController.java

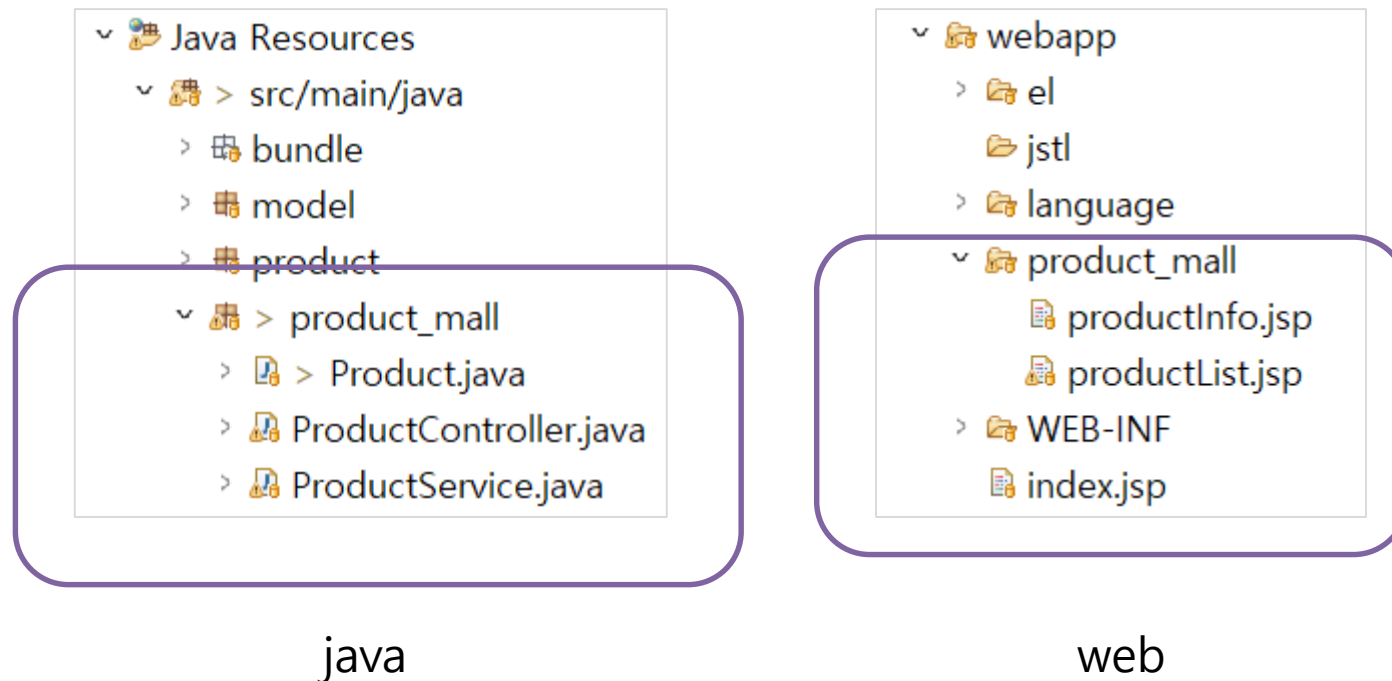
✓ 모델 구현은 DB를 사용하지 않고 자료 구조(Map)를 사용



상품 관리 프로젝트

➤ 상품 관리 프로젝트 만들기 - MVC 패턴

프로젝트 계층 구조



상품 관리 프로젝트

➤ 상품 관리

상품 목록

번호	상품명	제조사	가격
1	iPhone 17	애플(Apple)	1200000
2	LG Gram	LG 전자	2000000
3	Galaxy S25	삼성 전자	1500000



상품 정보

- 상품 코드: p101
- 상품 이름: Galaxy S25
- 제조사: 삼성 전자
- 가격: 1500000
- 등록일: 2025. 2. 25

[상품 목록으로](#)

상품 관리 프로젝트

➤ Product DTO 생성

```
public class Product {                                     product.java
    private String pid;      //상품번호
    private String pname;    //상품명
    private String maker;    //제조사
    private int price;       //가격
    private String date;     //등록일

    //생성자
    public Product(String pid, String pname, String maker,
                    int price, String date) {
        this.pid = pid;
        this.pname = pname;
        this.maker = maker;
        this.price = price;
        this.date = date;
    }
}
```



상품 관리 프로젝트

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



상품 관리 프로젝트

➤ 모델(Model) 역할

```
public class ProductService {
```

ProductService.java

```
    Map<String, Product> products = new HashMap<>();
```

```
    public ProductService() {
```

```
        //Product 객체 생성
```

```
        Product p1 = new Product("11", "Galaxy21", "삼성 전자",  
                                   1000000, "2023. 03. 16");
```

```
        Product p2 = new Product("12", "LG 그램", "LG 전자",  
                                   1400000, "2023. 04. 16");
```

```
        Product p3 = new Product("13", "iPhone", "Apple",  
                                   1200000, "2023. 05. 16");
```

```
        //Map에 저장
```

```
        products.put("11", p1);
```

```
        products.put("12", p2);
```

```
        products.put("13", p3);
```

```
    }
```



상품 관리 프로젝트

➤ 모델(Model) 역할

```
//전체 상품 목록 보기
public List<Product> getProductList(){
    return new ArrayList<>(products.values());
}

//제품 상세 내역
public Product getProduct(String pid) {
    return products.get(pid);
}
```


상품 관리 프로젝트

➤ Product 컨트롤러(서블릿)

ProductController.java

```
@WebServlet("/pcontrol") //http://localhost:8080/jweb02/pcontrol?action=list
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    ProductService service;

    public ProductController() {
        service = new ProductService();
    }

    //doGet(), doPost() 사용돼 됨
    protected void service(HttpServletRequest request, HttpServletResponse re
        //action 패턴
        String action = request.getParameter("action");
        String nextPage = "";
```



상품 관리 프로젝트

➤ Product 컨트롤러(서블릿)

```
if(action.equals("list")) { //상품 목록
    List<Product> productList = service.getProductList();
    request.setAttribute("productList", productList); //모델 생성
    nextPage = "/product_mall/productList.jsp"; //뷰 경로
}else if(action.equals("info")) {
    String pid = request.getParameter("pid");
    Product product = service.getProduct(pid);
    request.setAttribute("product", product);
    nextPage = "/product_mall/productInfo.jsp";
}
//포워딩
RequestDispatcher rd =
    request.getRequestDispatcher(nextPage);
rd.forward(request, response);
}
```

상품 관리 프로젝트

➤ 인덱스(index) 페이지

Good Mall에 오신 것을 환영합니다.

[상품 목록 보기](#)

index.jsp

```
<title>Good Mall입니다..</title>
</head>
<body>
    <h2>Good Mall에 오신 것을 환영합니다.</h2>

    <a href="/jweb02/pcontrol?action=list">상품 목록 보기</a>
</body>
```

상품 관리 프로젝트

➤ 상품 목록

```
<%@ taglib uri="jakarta.tags.core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>상품 목록</title>
  <style>
    body{margin: 20px;}
    table{
      border: 1px solid #ccc;
      border-collapse: collapse;
    }
    table th, td{
      border: 1px solid #ccc;
      padding: 5px 10px;
    }
  </style>
</head>
```

productList.jsp



상품 관리 프로젝트

➤ 상품 목록

```
<h2>상품 목록</h2>
<hr>
<table>
  <thead>
    <tr>
      <th>번호</th><th>상품명</th><th>제조사</th><th>가격</th>
    </tr>
  </thead>
  <tbody>
    <c:forEach var="product" items="${productList}" varStatus="i">
      <tr>
        <td>${i.count}</td>
        <td>
          <a href="/jweb02/pcontrol?action=info&pid=${product.pid}">
            ${product.pname}
          </a>
        </td>
        <td>${product.maker}</td>
        <td>${product.price}</td>
      </tr>
    </c:forEach>
  </tbody>
</table>
```

productList.jsp



상품 관리 프로젝트

➤ 상품 상세

```
<%@ taglib uri="jakarta.tags.core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>제품 상세 보기</title>
</head>
<body>
    <h2>상품 정보</h2>
    <hr>
    <ul>
        <li>상품 코드: ${product.pid}</li>
        <li>상품 이름: ${product.pname}</li>
        <li>제조사: ${product.maker}</li>
        <li>가격: ${product.price}</li>
        <li>등록일: ${product.date}</li>
    </ul>

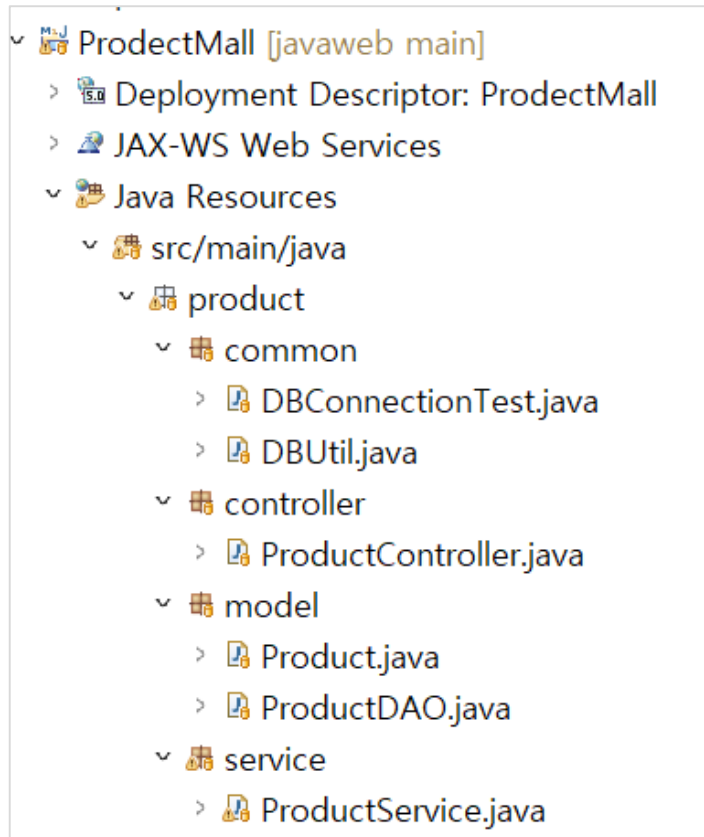
    <a href="/jweb02/pcontrol?action=list">상품 목록으로</a>
</body>
</html>
```

productInfo.jsp

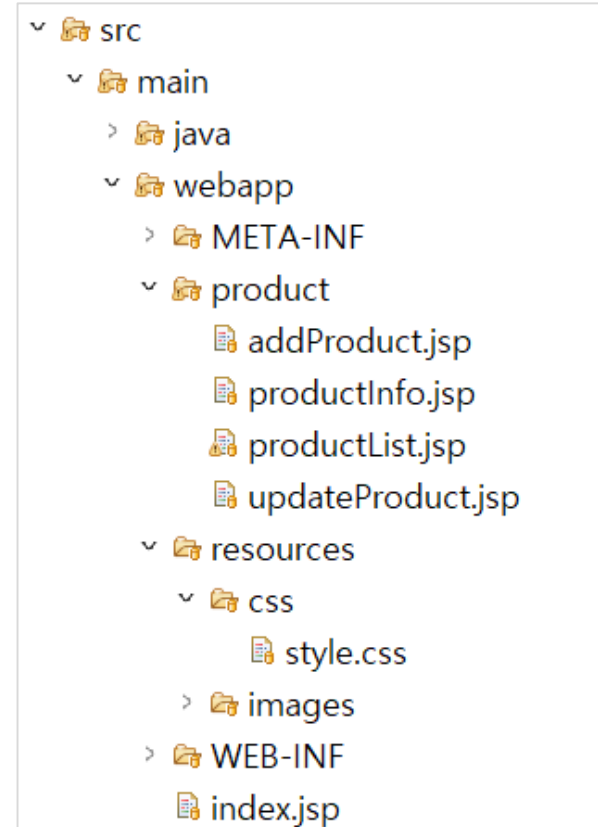


상품 관리 – DB 연동

- 상품관리 프로젝트 - 데이터베이스와 연동
- 프로젝트 이름 : ProductMall



java

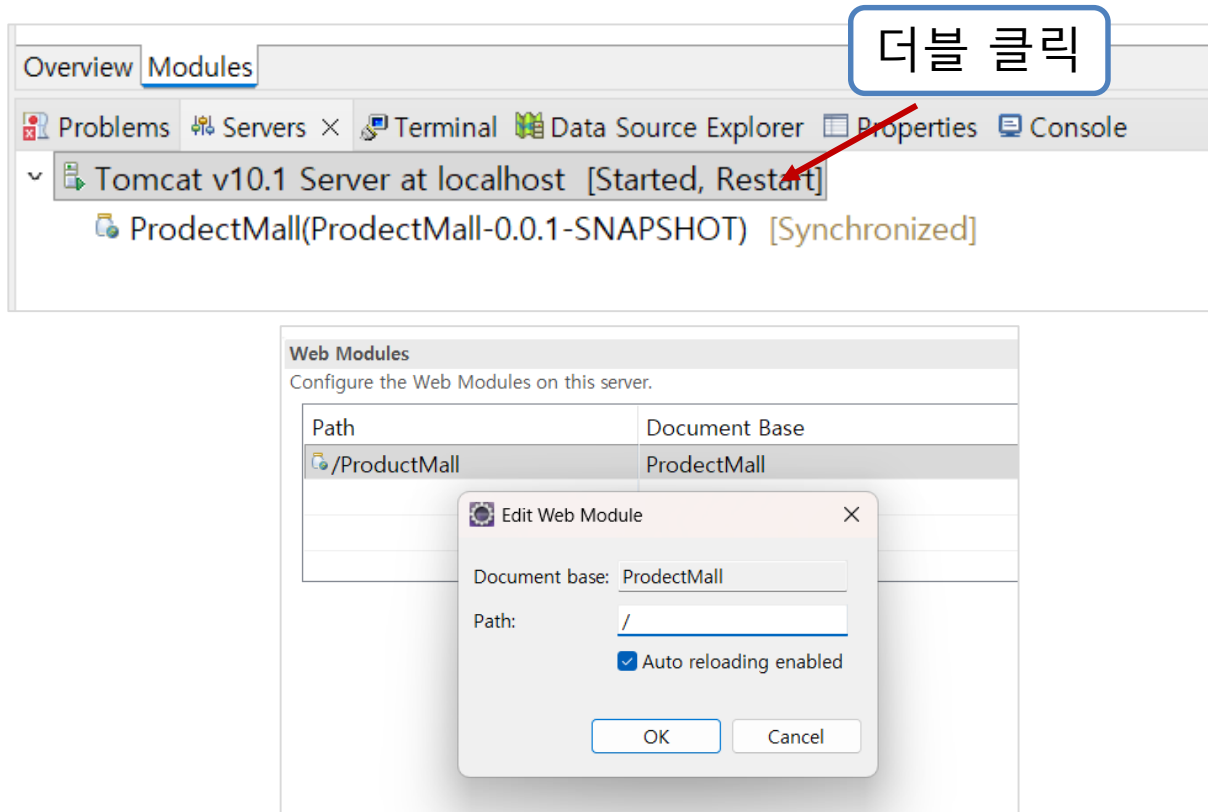


web



상품 관리 – DB 연동

- Tomcat Server localhost 경로 변경
- `http://localhost:8080/ProdictMall/` -> `http://localhost:8080/`



상품 관리 – DB 연동

- index 페이지

Good Mall에 오신 것을 환영합니다.



상품 목록

상품 관리 – DB 연동

▪ index 페이지

```
<head>
  <meta charset="UTF-8">
  <title>Welcome~ Good Mall...</title>
  <link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
  <section id="container">
    <div id="main">
      <h2>Good Mall에 오신 것을 환영합니다.</h2>

      <div class="main-pic">
        
      </div>

      <p><a href="/product?action=list">상품 목록</a></p>
    </div>
  </section>
</body>
</html>
```



상품 관리 – DB 연동

■ css 파일

```
#container{
    width: 80%;
    margin: 20px auto;
}
#main{text-align: center;}
a{text-decoration: none;}
a:hover{text-decoration: underline; }
ul li{margin: 10px;}
table{
    border: 1px solid #ccc;
    border-collapse: collapse;
}
h2{
    border-bottom: 2px solid #ccc;
    padding-bottom: 10px;
}
.main-pic img{
    width: 40%;
    border-radius: 10px;
}
```

```
table th, td{
    border: 1px solid #ccc;
    padding: 5px 10px;
}
form ul li{list-style: none;}
form ul li label{
    width: 80px;
    float: left;
}
```



상품 관리 – DB 연동

▪ JDBC - 데이터베이스 연동

- Maven Repository (메이븐 저장소) <https://mvnrepository.com/>
- mysql driver로 검색 > Connector/j

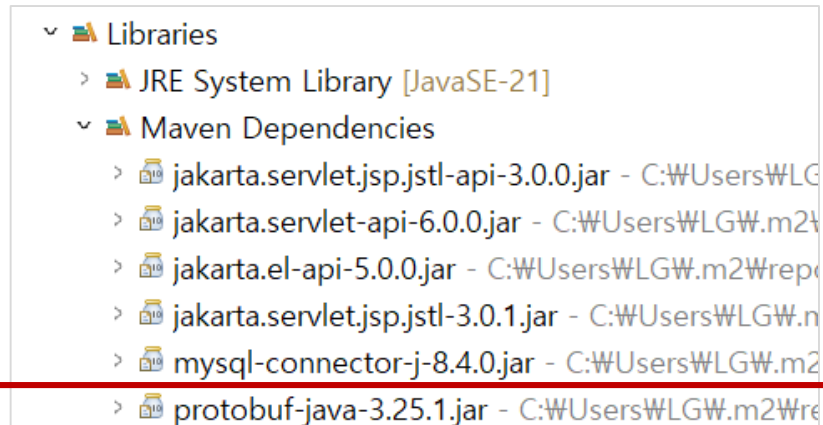


상품 관리 – DB 연동

▪ JDBC - 데이터베이스 연동

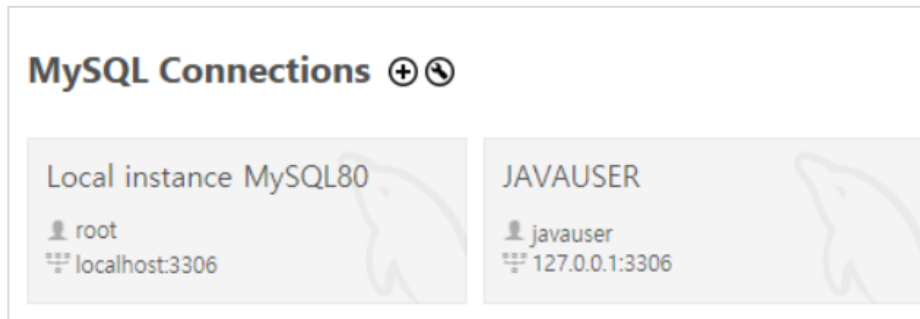
- pom.xml에 <dependency> 의존성 추가

```
<!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->  
<dependency>  
  <groupId>com.mysql</groupId>  
  <artifactId>mysql-connector-j</artifactId>  
  <version>8.4.0</version>  
</dependency>
```



상품 관리 – DB 연동

- MySQL WorkBench
- JAVAUSER 계정 만들고, 접속 화면 만들기



```
-- javauser 생성
create user javauser@localhost identified by 'pwjava';

-- 권한 부여하기
grant all privileges on *.* to javauser@localhost;

-- jspdb 생성
drop database jspdb;
```

상품 관리 – DB 연동

- jspdb > product 테이블 만들기

```
-- jspdb 생성
create database jspdb;

-- product 테이블 생성
create table if not exists product(
    pid varchar(10) primary key,
    pname text,
    maker varchar(20),
    price int,
    regdate date
);

insert into product
values ('p101', 'Galaxy S25', '삼성 전자', 1500000, '2025-02-25');
```

상품 관리 – DB 연동

▪ JDBC 연결 테스트

```
public class DBConnectionTest {  
    private static final String URL = "jdbc:mysql://localhost:3306/jspdb";  
    private static final String USER = "javauser";  
    private static final String PASSWORD = "pwjava";  
  
    public static void main(String[] args) {  
  
        try{  
            Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);  
            System.out.println(conn + "DB 접속 성공!!");  
        }catch(Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
com.mysql.cj.jdbc.ConnectionImpl@58ea606cDB 접속 성공!!
```



상품 관리 – DB 연동

▪ DBUtil 클래스

```
public class DBUtil {  
    private static final String URL = "jdbc:mysql://localhost:3306/jspdb";  
    private static final String USER = "javauser";  
    private static final String PASSWORD = "pwjava";  
  
    public static Connection getConnection() {  
  
        try{  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            return DriverManager.getConnection(URL, USER, PASSWORD);  
        }catch(Exception e) {  
            e.printStackTrace();  
            return null;  
        }  
    }  
}
```



상품 관리 – DB 연동

■ Product DTO 생성

```
public class Product {  
    private String pid;  
    private String pname;  
    private String maker;  
    private int price;  
    private String date;  
  
    //생성자  
    public Product(String pid, String pname,  
        String maker, int price, String date) {  
        super();  
        this.pid = pid;  
        this.pname = pname;  
        this.maker = maker;  
        this.price = price;  
        this.date = date;  
    }  
}
```

product.java



상품 관리 – DB 연동

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



상품 관리 – DB 연동

▪ Product DAO – 상품 등록 메서드

```
public class ProductDAO {  
    //상품 등록  
    public void addProduct(Product product) {  
        String sql = "insert into product (pid, pname, maker, price, regdate) "  
            + "values (?, ?, ?, ?, ?)";  
  
        try(Connection conn = DBUtil.getConnection();  
            PreparedStatement pstmt = conn.prepareStatement(sql)){  
            pstmt.setString(1, product.getPid());  
            pstmt.setString(2, product.getPname());  
            pstmt.setString(3, product.getMaker());  
            pstmt.setInt(4, product.getPrice());  
            pstmt.setString(5, product.getRegdate());  
            pstmt.executeUpdate();  
        }catch(Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

상품 관리 – DB 연동

▪ Product DAO – 상품 목록 메서드

```
public List<Product> getProductList(){
    List<Product> list = new ArrayList<>();
    String sql = "select * from product";

    try(Connection conn = DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery()){

        while(rs.next()) {
            Product product = new Product(
                rs.getString("pid"),
                rs.getString("pname"),
                rs.getString("maker"),
                rs.getInt("price"),
                rs.getString("regdate")
            );
            list.add(product);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return list;
}
```



상품 관리 – DB 연동

■ Product DAO – 상품 상세 보기 메서드

```
public Product getProduct(String pid) {  
    Product product = null;  
    String sql = "select * from product where pid = ?";  
  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement pstmt = conn.prepareStatement(sql)){  
  
        pstmt.setString(1, pid);  
        ResultSet rs = pstmt.executeQuery();  
  
        if(rs.next()) {  
            product = new Product(  
                rs.getString("pid"),  
                rs.getString("pname"),  
                rs.getString("maker"),  
                rs.getInt("price"),  
                rs.getString("regdate")  
            );  
        }  
    } catch(Exception e) {  
        e.printStackTrace();  
    }  
    return product;  
}
```



상품 관리 – DB 연동

- Product DAO – 상품 삭제 메서드

```
public void deleteProduct(String pid) {  
    String sql = "delete from product where pid = ?";  
  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement pstmt = conn.prepareStatement(sql)){  
        pstmt.setString(1, pid);  
        pstmt.executeUpdate();  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```

상품 관리 – DB 연동

▪ Product DAO – 상품 수정 메서드

```
public void updateProduct(Product product) {  
    String sql = "update product set pname=?, maker=?, price=?, regdate=? "  
        + "where pid = ?";  
  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement pstmt = conn.prepareStatement(sql)){  
        pstmt.setString(1, product.getPname());  
        pstmt.setString(2, product.getMaker());  
        pstmt.setInt(3, product.getPrice());  
        pstmt.setString(4, product.getRegdate());  
        pstmt.setString(5, product.getPid());  
        pstmt.executeUpdate();  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```


상품 관리 – DB 연동

▪ ProductService – 서비스 계층

```
public class ProductService {  
  
    ProductDAO dao = new ProductDAO();  
  
    //상품 등록  
    public void addProduct(Product product) {  
        dao.addProduct(product);  
    }  
  
    //상품 목록  
    public List<Product> getProductList(){  
        return dao.getProductList();  
    }  
  
    //상품 상세  
    public Product getProduct(String pid) {  
        return dao.getProduct(pid);  
    }  
}
```



상품 관리 – DB 연동

- ProductService – 서비스 계층

```
//상품 삭제
public void deleteProduct(String pid) {
    dao.deleteProduct(pid);
}

//상품 수정
public void updateProduct(Product product) {
    dao.updateProduct(product);
}
}
```

상품 관리 – DB 연동

▪ ProductController – 서블릿 컨트롤러 계층

```
@WebServlet("/product")
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    ProductService service;

    public ProductController() {
        service = new ProductService();
    }

    protected void service(HttpServletRequest request, HttpServletResponse response) {
        //action 패턴
        String action = request.getParameter("action");
        String nextPage = "";

        if(action.equals("list")) { //목록 보기
            List<Product> productList = service.getProductList();
            request.setAttribute("productList", productList);
            nextPage = "/product/productList.jsp";
        }
    }
}
```



상품 관리 – DB 연동

▪ ProductController – 서블릿 컨트롤러 계층

```
}else if(action.equals("info")) { //상세 보기
    String pid = request.getParameter("pid");
    Product product = service.getProduct(pid);
    request.setAttribute("product", product);
    nextPage = "/product/productInfo.jsp";
}else if(action.equals("addForm")) { //등록 폼
    nextPage = "/product/addProduct.jsp";
}else if(action.equals("add")) { //등록 처리
    String pid = request.getParameter("pid");
    String pname = request.getParameter("pname");
    String maker = request.getParameter("maker");
    int price = Integer.parseInt(request.getParameter("price"));
    String regdate = request.getParameter("regdate");

    Product product = new Product(pid, pname, maker, price, regdate);
    service.addProduct(product);

    //등록후 목록 페이지로 이동
    response.sendRedirect("/product?action=list");
    return; //즉시 종료
```



상품 관리 – DB 연동

▪ ProductController – 서블릿 컨트롤러 계층

```
}else if(action.equals("updateForm")) { //수정 페이지
    String pid = request.getParameter("pid");
    Product product = service.getProduct(pid);
    request.setAttribute("product", product);
    nextPage = "/product/updateProduct.jsp";
}else if(action.equals("update")) { //수정 처리
    String pid = request.getParameter("pid");
    String pname = request.getParameter("pname");
    String maker = request.getParameter("maker");
    int price = Integer.parseInt(request.getParameter("price"));
    String regdate = request.getParameter("regdate");

    Product product = new Product(pid, pname, maker, price, regdate);
    service.updateProduct(product);
    //수정후 상세 페이지로 이동
    response.sendRedirect("/product?action=list&pid=" + pid);
    return;
}

RequestDispatcher rd =
    request.getRequestDispatcher(nextPage);
rd.forward(request, response);
}
```



상품 관리 – DB 연동

- 상품 목록 페이지

상품 목록

번호	상품명	제조사	가격	등록일
1	Galaxy S25	삼성 전자	1500000	2025-02-25
2	iPhone 17	애플(Apple)	1200000	2025-09-19

상품 등록

상품 관리 – DB 연동

- 상품 목록 페이지

```
<head>
  <meta charset="UTF-8">
  <title>상품 목록</title>
  <link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
  <section id="container">
    <h2>상품 목록</h2>
    <table>
      <thead>
        <tr>
          <th>번호</th>
          <th>상품명</th>
          <th>제조사</th>
          <th>가격</th>
          <th>등록일</th>
        </tr>
      </thead>
    </table>
  </section>
</body>
```

productList.jsp



상품 관리 – DB 연동

■ 상품 목록 페이지

```
<tbody>
  <c:forEach var="product" items="${productList}" varStatus="i">
    <tr>
      <td>${i.count}</td>
      <td>
        <a href="/product?action=info&pid=${product.pid}">
          ${product.pname}
        </a>
      </td>
      <td>${product.maker}</td>
      <td>${product.price}</td>
      <td>${product.regdate}</td>
    </tr>
  </c:forEach>
</tbody>
</table>
<p><a href="/product?action=addForm">상품 등록</a></p>
</section>
```



상품 관리 – DB 연동

■ 상품 등록

상품 등록

상품 코드	<input type="text" value="p102"/>
상품명	<input type="text" value="삼성노트북"/>
제조사	<input type="text" value="삼성전자"/>
가격	<input type="text" value="1600000"/>
등록일	<input type="text" value="2025-10-12"/> <input type="button" value="📅"/>

상품 관리 – DB 연동

■ 상품 등록

```
<section id="container">
  <h2>상품 등록</h2>
  <form action="/product?action=add" method="post">
    <ul>
      <li>
        <label>상품 코드</label>
        <input type="text" name="pid">
      </li>
      <li>
        <label>상품명</label>
        <input type="text" name="pname">
      </li>
      <li>
        <label>제조사</label>
        <input type="text" name="maker">
      </li>
    </ul>
  </form>
</section>
```

addProduct.jsp



상품 관리 – DB 연동

■ 상품 등록

```
        <li>
            <label>가격</label>
            <input type="number" name="price">
        </li>
        <li>
            <label>등록일</label>
            <input type="date" name="regdate">
        </li>
    </ul>
    <p>
        <input type="submit" value="등록">
        <input type="reset" value="취소">
        <a href="/product?action=list">
            <button type="button">목록</button>
        </a>
    </p>
</form>
</section>
```



상품 관리 – DB 연동

■ 상품 정보

상품 정보

- 상품 코드: p101
- 상품 이름: Galaxy S25
- 제조사: 삼성 전자
- 가격: 1500000
- 등록일: 2025-02-25

[수정](#)[삭제](#)[목록](#)

상품 관리 – DB 연동

■ 상품 정보

```
<h2>상품 정보</h2>
<ul>
  <li>상품 코드: ${product.pid}</li>
  <li>상품 이름: ${product.pname}</li>
  <li>제조사: ${product.maker}</li>
  <li>가격: ${product.price}</li>
  <li>등록일: ${product.regdate}</li>
</ul>
<div>
  <a href="/product?action=updateForm&pid=${product.pid}">
    <button type="submit">수정</button>
  </a>
  <a onclick="return confirm('정말로 삭제하시겠습니까?')"
    href="/product?action=delete&pid=${product.pid}">
    <button type="submit">삭제</button></a>
  <a href="/product?action=list">
    <button type="button">목록</button>
  </a>
</div>
```

productInfo.jsp



상품 관리 – DB 연동

■ 상품 삭제

상품 정보

- 상품 코드: p
- 상품 이름: 삼성
- 제조사: 삼성전자
- 가격: 1700000
- 등록일: 2025-09-18

수정삭제목록

localhost:8080 내용:

정말로 삭제하시겠습니까?

확인취소

상품 관리 – DB 연동

■ 상품 수정

상품 수정

상품 코드	<input type="text" value="p102"/>
상품명	<input type="text" value="Galaxy Book5"/>
제조사	<input type="text" value="삼성전자"/>
가격	<input type="text" value="1900000"/>
등록일	<input type="text" value="2025-10-12"/> 

상품 관리 – DB 연동

■ 상품 수정

updateProduct.jsp

```
<h2>상품 수정</h2>
<form action="/product?action=update" method="post">
  <ul>
    <li>
      <label>상품 코드</label>
      <input type="text" name="pid" value="${product.pid}">
    </li>
    <li>
      <label>상품명</label>
      <input type="text" name="pname" value="${product.pname}">
    </li>
    <li>
      <label>제조사</label>
      <input type="text" name="maker" value="${product.maker}">
    </li>
  </ul>
</form>
```


상품 관리 – DB 연동

■ 상품 수정

```
<li>
  <label>가격</label>
  <input type="number" name="price" value="${product.price}">
</li>
<li>
  <label>등록일</label>
  <input type="date" name="regdate" value="${product.regdate}">
</li>
</ul>
<p>
  <input type="submit" value="수정">
  <input type="reset" value="취소">
  <a href="/product?action=list">
    <button type="button">목록</button>
  </a>
</p>
</form>
```