

JSP 웹 프로그래밍

내장객체

내장 객체(implicit object)

- ▶ JSP 페이지에서 사용할 수 있도록 JSP 컨테이너에 미리 정의된 객체
- ▶ JSP 페이지가 서블릿 프로그램으로 번역될 때 JSP 컨테이너가 자동으로 내장 객체를 멤버 변수, 메소드 매개변수 등의 각종 참조 변수(객체)로 포함
- ▶ JSP 페이지에 별도의 import 문 없이 자유롭게 사용 가능
- ▶ 스크립틀릿 태그나 표현문 태그에 선언을 하거나 객체를 생성하지 않고도 직접 호출하여 사용 가능



request 내장 객체

- ▶ JSP 페이지에서 가장 많이 사용되는 기본 내장 객체
- ▶ 웹 브라우저에서 서버의 JSP 페이지로 전달하는 정보를 저장
 - ▶ 폼 페이지로부터 입력된 데이터를 전달하는 요청 파라미터 값을 JSP 페이지로 가져옴
- ▶ JSP 컨테이너는 웹 브라우저에서 서버로 전달되는 정보를 처리하기 위해
javax.servlet.http.HttpServletRequest 객체 타입의 request 내장 객체를 사용하여 사용자의 요구 사항을 얻어냄



request 내장 객체: 요청 파라미터

▶ 요청 파라미터

- ▶ 사용자가 폼 페이지에 데이터를 입력한 후 서버에 전송할 때 전달되는 폼 페이지의 입력된 정보 형태
- ▶ <name=value> 형식으로 웹 브라우저에서 서버의 JSP 페이지로 전송

▶ 요청 파라미터 관련 메소드의 종류

요청 파라미터 관련 메소드	반환 유형	설명
getParameter(String name)	String	요청 파라미터 이름이 name인 값을 전달받습니다. 요청 파라미터 값이 없으면 null을 반환합니다.
getParameterValues(String name)	String[]	모든 요청 파라미터 이름이 name인 값을 배열 형태로 전달받습니다. 요청 파라미터 값이 없으면 null을 반환합니다.
getParameterNames()	java.util.Enumeration	모든 요청 파라미터의 이름과 값을 Enumeration 객체 타입으로 전달받습니다.
getParameterMap()	java.util.Map	모든 요청 파라미터의 이름과 값을 Map 객체 타입으로 전달받습니다[Map 객체 타입은 (요청 파라미터 이름, 값) 형식으로 구성됩니다]

request 내장 객체의 사용 예제

- ▶ Request내장 객체로 폼 페이지로부터 아이디와 비밀번호를 전송 받아 출력하기

```
request01.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Implicit Objects</title>
5 </head>
6 <body>
7 <form action="request01_process.jsp" method="post">
8     <p>
9         아 이 디 : <input type="text" name="id">
10    <p>
11        비밀번호:<input type="text" name="passwd">
12    <p>
13        <input type="submit" value="전송">
14    </form>
15 </body>
16 </html>
```

request 내장 객체의 사용 예제

request01_process.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Implicit Objects</title>
5 </head>
6 <body>
7 <%
8     request.setCharacterEncoding("utf-8");
9     String userId = request.getParameter("id");
10    String password = request.getParameter("passwd");
11 %>
12 <p>
13     아이디 :
14     <%=userId%>
15 <p>
16     비밀번호 :
17     <%=password%>
18 </body>
19 </html>
```

http://localhost:8090/JSP

아이디 :

비밀번호:

전송

http://localhost:80

아이디 : 관리자

비밀번호 : 1111

request 내장 객체의 기능

▶ 요청 HTTP 헤더 관련 메소드

- ▶ 웹 브라우저는 HTTP 헤더에 추가적인 정보를 담아 서버로 전송

요청 HTTP 헤더 관련 메소드	반환 유형	설명
getHeader(String name)	String	설정된 name의 헤더 값을 가져옵니다.
getHeaders(String name)	Enumeration	설정된 name의 헤더 목록 값을 가져옵니다.
getHeaderNames()	Enumeration	모든 헤더 이름을 가져옵니다.
getIntHeader(String name)	int	설정된 name의 헤더 값을 정수로 가져옵니다.
getDateHeader(String name)	long	설정된 name의 헤더 값을 시간 값으로 가져옵니다.
getCookies()	javax.servlet.http.Cookie	모든 쿠키 값을 가져옵니다.



request 내장 객체의 사용 예제

- ▶ request 내장 객체로 모든 HTTP헤더 정보 값 출력하기

```
request02.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.util.Enumeration"%>
3 <html>
4 <head>
5 <title>Implicit Objects</title>
6 </head>
7 <body>
8 <%
9     Enumeration<String> en = request.getHeaderNames();
10    while (en.hasMoreElements()) {
11        String headerName = en.nextElement();
12        String headerValue = request.getHeader(headerName);
13    %>
14    <%=headerName%>
15    :
16    <%=headerValue%><br>
17 <%
18     }
19 %>
20 </body>
21 </html>
```

http://localhost:8090/JSPTest/Chapter05/request02.jsp

accept : image/gif, image/jpeg, image/pjpeg, application/x-ms-application, application/xhtml+xml, application/x-ms-xbap, */*
accept-language : ko-KR
cache-control : no-cache
ua-cpu : AMD64
accept-encoding : gzip, deflate
user-agent : Mozilla/5.0 (Windows NT 6.2; Win64; x64; Trident/7.0; rv:11.0) like Gecko
host : localhost:8090
connection : Keep-Alive

request 내장 객체의 기능

▶ 웹 브라우저/서버 관련 메소드

웹 브라우저/서버 관련 메소드	반환 유형	설명
getRemoteAddr()	String	웹 브라우저의 IP 주소를 가져옵니다.
getContentLength()	long	웹 브라우저의 요청 파라미터 길이를 가져옵니다.
getCharacterEncoding()	String	웹 브라우저의 문자 인코딩을 가져옵니다.
getContentType()	String	웹 브라우저의 콘텐츠 유형을 가져옵니다.
getProtocol()	String	웹 브라우저의 요청 프로토콜을 가져옵니다.
getMethod()	String	웹 브라우저의 HTTP 요청 메소드(GET, POST)를 가져옵니다.
getRequestURI()	String	웹 브라우저가 요청한 URI 경로를 가져옵니다.
getContextPath()	String	현재 JSP 페이지의 웹 애플리케이션 컨텍스트 경로를 가져옵니다.
getServerName()	String	서버 이름을 가져옵니다.
getServerPort()	int	실행 중인 서버 포트 번호를 가져옵니다.
getQueryString()	String	웹 브라우저의 전체 요청 파라미터 문자열(물음표(?) 다음 URL에 할당된 문자열)을 가져옵니다.

request 내장 객체의 사용 예제

- ▶ request 내장 객체로 모든 웹 브라우저 및 서버 정보 값 출력하기

request03.jsp

```
1 <%@ page contentType="text/html; charset=utf-8" %>
2 <html>
3 <head>
4 <head>
5   <title> Implicit Objects </title>
6 </head>
7 <body>
8   <p>클라이언트 IP : <%=request.getRemoteAddr() %>
9   <p>요청 정보 길이 : <%=request.getContentLength() %>
10  <p>요청 정보 인코딩 : <%=request.getCharacterEncoding() %>
11  <p>요청 정보 콘텐츠 유형 : <%=request.getContentType() %>
12  <p>요청 정보 프로토콜 : <%=request.getProtocol() %>
13  <p>요청 정보 전송방식 : <%=request.getMethod() %>
14  <p>요청 URI : <%=request.getRequestURI() %>
15  <p>컨텍스트 경로 : <%=request.getContextPath() %>
16  <p>서버이름 : <%=request.getServerName() %>
17  <p>서버포트 : <%=request.getServerPort() %>
18  <p>쿼리문 : <%=request.getQueryString() %>
19 </body>
20 </html>
```

http://localhost:8090/JSPTest/Chapter05/

클라이언트 IP : 0:0:0:0:0:0:1
요청 정보 길이 : -1
요청 정보 인코딩 : null
요청 정보 콘텐츠 유형 : null
요청 정보 프로토콜 : HTTP/1.1
요청 정보 전송방식 : GET
요청 URI : /JSPTest/Chapter05/request03.jsp
컨텍스트 경로 : /JSPTest
서버이름 : localhost
서버포트 : 8090
쿼리문 : null

response 내장 객체

- ▶ 사용자의 요청을 처리한 결과를 서버에서 웹 브라우저로 전달하는 정보를 저장
- ▶ 서버는 응답 헤더와 요청 처리 결과 데이터를 웹 브라우저로 보냄.
- ▶ JSP 컨테이너는 서버에서 웹 브라우저로 응답하는 정보를 처리하기 위해
javax.servlet.http.HttpServletResponse 객체 타입의 response 내장 객체를 사용하여 사용자의 요청에 응답



response 내장 객체의 기능

▶ 페이지 이동 관련 메소드

- ▶ 페이지 이동 = 리다이렉션(redirection)
 - ▶ 사용자가 새로운 페이지를 요청할 때와 같이 페이지를 강제로 이동하는 것
- ▶ 서버는 웹 브라우저에 다른 페이지로 강제 이동하도록 response 내장 객체의 리다이렉션 메소드를 제공
- ▶ 페이지 이동 시에는 문자 인코딩을 알맞게 설정해야 함

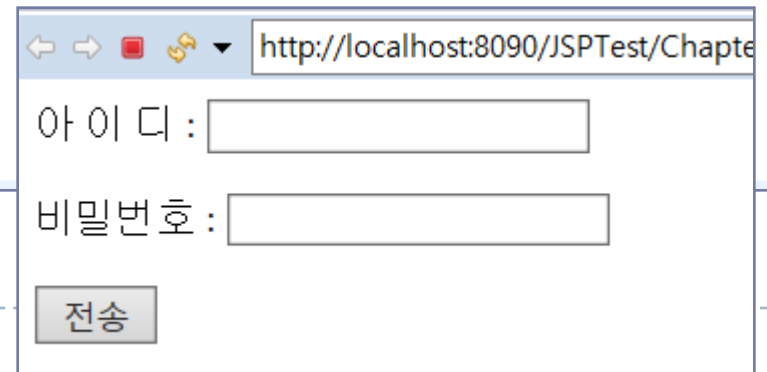
▶ 페이지 이동 관련 메소드의 종류

페이지 이동 관련 메소드	반환 유형	설명
sendRedirect(String url)	void	설정된 URL 페이지로 강제 이동합니다.



request내장 객체로 페이지 이동하기 예제

```
*response01.jsp
1 <%@ page contentType="text/html; charset=utf-8" %>
2 <html>
3 <head>
4 <head>
5     <title> Implicit Objects </title>
6 </head>
7 <body>
8     <form action="response01_process.jsp" method="post">
9         <p>아 이 디 : <input type="text" name="id">
10        <p>비밀번호 : <input type="text" name="passwd">
11        <p><input type="submit" value="전송">
12    </form>
13 </body>
14 </html>
```



http://localhost:8090/JSPTest/Chapte

아 이 디 :

비밀번호 :

request내장 객체로 페이지 이동하기 예제

response01_process.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Implicit Objects</title>
5 </head>
6 <body>
7     <%
8         request.setCharacterEncoding("utf-8");
9         String userId = request.getParameter("id");
10        String password = request.getParameter("passwd");
11        if(userId.equals("관리자") && password.equals("1234")){
12            response.sendRedirect("response01_success.jsp");
13        }else{
14            response.sendRedirect("response01_failed.jsp");
15        }
16    %>
17 </body>
18 </html>
```

request내장 객체로 페이지 이동하기 예제

response01_success.jsp

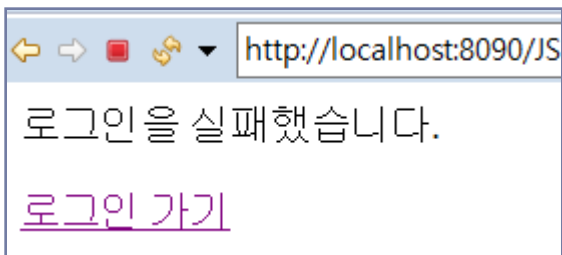
```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Implicit Objects</title>
5 </head>
6 <body>
7     로그인을 성공했습니다!!
8 </body>
9 </html>
```

← → ⏏ 🔄 ▼ http://localhost:8090/JSP

로그인을 성공했습니다!!

request내장 객체로 페이지 이동하기 예제

```
response01_failed.jsp ✕  
1 <%@ page contentType="text/html; charset=utf-8"%>  
2 <html>  
3 <head>  
4 <title>Implicit Objects</title>  
5 </head>  
6 <body>  
7     로그인을 실패했습니다.  
8     <p> <a href="./response01.jsp">로그인 가기</a>  
9 </body>  
10 </html>
```



response 내장 객체의 기능

▶ 응답 HTTP 헤더 관련 메소드

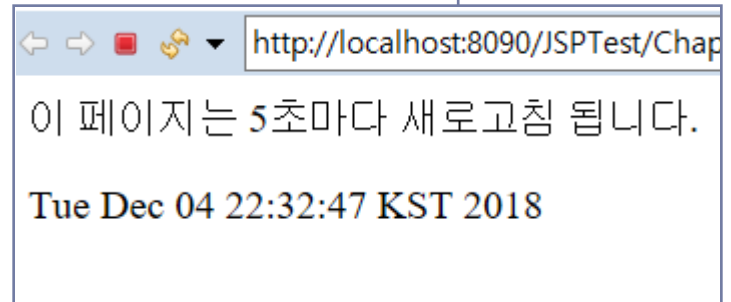
- ▶ 서버가 웹 브라우저에 응답하는 정보에 헤더를 추가하는 기능을 제공
- ▶ 헤더 정보에는 주로 서버에 대한 정보가 저장되어 있음
- ▶ 응답 HTTP 헤더 관련 메소드의 종류

응답 HTTP 헤더 관련 메소드	반환 유형	설명
<code>addCookie(Cookie cookie)</code>	<code>void</code>	쿠키를 추가합니다.
<code>addDateHeader(String name, long date)</code>	<code>void</code>	설정된 헤더 이름 name에 날짜/시간을 추가합니다.
<code>addHeader(String name, String value)</code>	<code>void</code>	설정된 헤더 이름 name에 value를 추가합니다.
<code>addIntHeader(String name, int value)</code>	<code>void</code>	설정된 헤더 이름 name에 정수값 value를 추가합니다.
<code>setDateHeader(String name, long date)</code>	<code>void</code>	설정된 헤더 이름 name에 날짜/시간을 설정합니다.
<code>setHeader(String name, String value)</code>	<code>void</code>	설정된 헤더 이름 name에 문자열 값 value를 설정합니다.
<code>setIntHeader(String name, int value)</code>	<code>void</code>	설정된 헤더 이름 name에 정수값 value를 설정합니다.
<code>containsHeader(String name)</code>	<code>boolean</code>	설정된 헤더 이름 name이 HTTP 헤더에 포함되었는지 여부를 확인합니다. 포함하고 있는 경우 <code>true</code> 를 반환하고, 그렇지 않은 경우 <code>false</code> 를 반환합니다.
<code>getHeader(String name)</code>	<code>String</code>	설정된 헤더 이름 name 값을 가져옵니다.

response 내장 객체 사용 예제

▶ response 내장 객체로 5초마다 JSP페이지 갱신하기

```
response02.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <head>
5 <title>Implicit Objects</title>
6 </head>
7 <body>
8     <p>
9         이 페이지는 5초마다 새로고침 됩니다.
10     <%
11         response.setIntHeader("Refresh", 5);
12     %>
13     <p><%=new java.util.Date()%>
14 </body>
15 </html>
```



response 내장 객체의 기능

▶ 응답 콘텐츠 관련 메소드

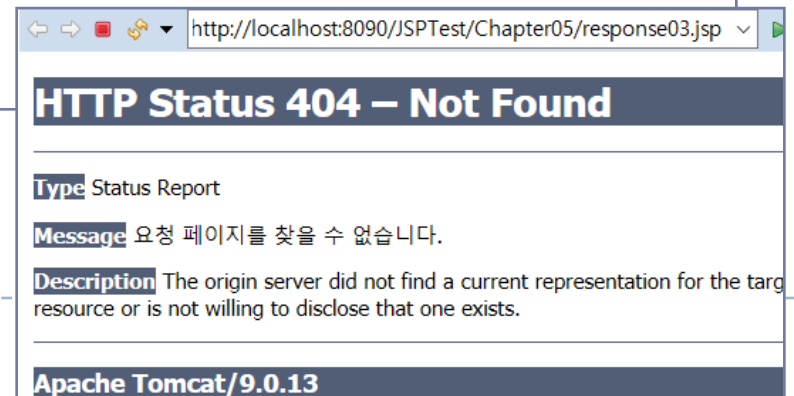
- ▶ response 내장 객체는 웹 브라우저로 응답하기 위해 MIME 유형, 문자 인코딩, 오류 메시지, 상태 코드 등을 설정하고 가져오는 응답 콘텐츠 관련 메소드를 제공
- ▶ 응답 콘텐츠 관련 메소드의 종류

응답 콘텐츠 관련 메소드	반환 유형	설명
setContentType(String type)	void	웹 브라우저에 응답할 MIME 유형을 설정합니다.
getContentType()	String	웹 브라우저에 응답할 MIME 유형을 가져옵니다.
setCharacterEncoding(String charset)	void	웹 브라우저에 응답할 문자 인코딩을 설정합니다.
getCharacterEncoding()	String	웹 브라우저에 응답할 문자 인코딩을 가져옵니다.
sendError(int status_code, String message)	void	웹 브라우저에 응답할 오류(코드 및 오류 메시지)를 설정합니다.
setStatus(int statuscode)	void	웹 브라우저에 응답할 HTTP 코드를 설정합니다.

response 내장 객체의 사용 예제

- ▶ response 내장 객체로 오류 응답 코드와 오류 메시지 보내기

```
response03.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <head>
5 <title>Implicit Objects</title>
6 </head>
7 <body>
8 <%
9     response.sendError(404, "요청 페이지를 찾을 수 없습니다.");
10 %>
11 </body>
12 </html>
```



out 내장 객체

- ▶ 웹 브라우저에 데이터를 전송하는 출력 스트림 객체
- ▶ JSP 컨테이너는 JSP 페이지에 사용되는 모든 표현문 태그와 HTML, 일반 텍스트 등을 out 내장 객체를 통해 웹 브라우저에 그대로 전달
- ▶ 스크립틀릿 태그에 사용하여 단순히 값을 출력하는 표현문 태그(<%= ...%>)와 같은 결과를 얻을 수 있음



out 내장 객체 메소드의 종류

out 내장 객체 메소드	반환 유형	설명
print(String str)	void	설정된 str 값을 웹 브라우저에 출력합니다.
println(String str)	void	설정된 str 값을 웹 브라우저에 출력합니다. 이때 줄 바꿈(\r\n 또는 \n)이 적용되지 않습니다.
newLine()	void	줄바꿈(\r\n 또는 \n)을 출력합니다.
getBufferSize()	int	현재 출력 버퍼의 크기를 가져옵니다.
getRemaining()	int	현재 남아 있는 출력 버퍼의 크기를 가져옵니다.
clear()	void	현재 출력 버퍼에 저장되어 있는 내용을 웹 브라우저에 전송하지 않고 비웁니다. 만약 버퍼가 이미 플러시되었다면 IOException이 발생합니다.
clearBuffer()	void	현재 출력 버퍼에 저장되어 있는 내용을 웹 브라우저에 전송하지 않고 비웁니다. 만약 버퍼가 이미 플러시되었다면 IOException이 발생하지 않습니다.
flush()	void	현재 출력 버퍼에 저장되어 있는 내용을 웹 브라우저에 전송하고 비웁니다.
isAutoFlush()	boolean	출력 버퍼가 채워졌을 때의 처리를 결정합니다. 자동으로 플러시하는 경우 true를 반환하고, 그렇지 않은 경우 false를 반환합니다.



out 내장 객체의 사용 예제

▶ out내장 객체로 오늘 날짜 및 시각 출력하기

```
out01.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <head>
5 <title>Implicit Objects</title>
6 </head>
7 <body>
8 <%
9     out.println("오늘의 날짜 및 시각 " + "<br>");
10    out.println(java.util.Calendar.getInstance().getTime());
11 %>
12 </body>
13 </html>
```

← → ⏏ 📌 http://localhost:8090/JSPTest/Cha

오늘의 날짜 및 시각
Tue Dec 04 22:46:21 KST 2018

out 내장 객체의 사용 예제

- ▶ out내장 객체로 폼 페이지에서 아이디와 비밀번호를 전송 받아 출력하기

out02.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Implicit Objects</title>
5 </head>
6 <body>
7     <form action="out02_process.jsp" method="post">
8         <p>
9             아 이 디 : <input type="text" name="id">
10        <p>
11            비밀번호:<input type="text" name="passwd">
12        <p>
13            <input type="submit" value="전송">
14    </form>
15 </body>
16 </html>
```

http://localhost:8090/JSPT

아 이 디 : 관리자

비밀번호: 1234

전송

상품 상세 정보 표시하기

[Home](#)

상품 정보

iPhone 6s

4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera

상품 코드 : **P1234**

제조사 : Apple

분류 : Smart Phone

재고 수 : 1000

800000원

[상품 주문 »](#)[상품 목록 »](#)

© WebMarket

[Home](#)

상품 목록

iPhone 6s

4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera

800000원

[상세 정보 »](#)

LG PC 그램

13.3-inch, IPS LED display, 5rd Generation Intel Core processors

1500000원

[상세 정보 »](#)

Galaxy Tab S

212.8*125.6*6.6mm, Super AMOLED display, Octa-Core processor

900000원

[상세 정보 »](#)

© WebMarket

상품 데이터 접근 클래스 만들기

▶ 상품 상세 정보를 가져오는 메소드 만들기

ProductRepository.java

```
1 package dao;
2 import java.util.ArrayList;
4 public class ProductRepository {
5     public Product getProductById(String productId)
6     {
7         Product productById = null;
8         for(int i=0;i<listOfProducts.size();i++)
9         {
10             Product product = listOfProducts.get(i);
11             if(product !=null && product.getProductId() !=null
12                && product.getProductId().equals(productId))
13             {
14                 productById =product;
15                 break;
16             }
17         }
18         return productById;
19     }
}
```

상품 상세 정보 버튼 만들기(부트스트랩 적용 전)

▶ 상품 목록에 링크 추가하기

```
<%  
    for (int i = 0; i < listOfProducts.size(); i++) {  
        Product product = listOfProducts.get(i);  
    %>  
<h3><%=product.getPname()%></h3>  
<p><%=product.getDescription()%>  
<p><%=product.getUnitPrice()%>원  
<p><a href="./product.jsp?id=<%=product.getProductId()%>">  
    상세 정보 &raquo;</a>  
<%  
    }  
%>
```

상품 목록

iPhone 6s

4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera

800000원

[상세 정보 »](#)

상품 상세 정보 페이지 만들기(부트스트랩 적용 전)

[Home](#)

상품 정보

iPhone 6s

4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera

상품 코드 : P1234

제조사 : Apple

분류 : Smart Phone

재고 수 : 1000

800000원

[상품 주문](#) » [상품 목록](#) »

© WebMarket

상품 상세 정보 페이지 만들기

```
product.jsp ✕
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="dto.Product"%>
3 <jsp:useBean id="productDAO" class="dao.ProductRepository"
4   scope="session" />
5 <html>
6 <head>
7 <title>상품 상세 정보</title>
8 </head>
9 <body>
10 <jsp:include page="menu.jsp" />
11 <h1>상품 정보</h1>
12 <%
13     String id = request.getParameter("id");
14     Product product = productDAO.getProductById(id);
15 %>
```

상품 상세 정보 페이지 만들기(product.jsp)

```
16 <h3><%=product.getPname()%></h3>
17 <p><%=product.getDescription()%>
18 <p><b>상품 코드 : </b><%=product.getProductId()%>
19 <p><b>제조사</b> : <%=product.getManufacturer()%>
20 <p><b>분류</b> : <%=product.getCategory()%>
21 <p><b>재고 수</b> : <%=product.getUnitsInStock()%>
22 <h4><%=product.getUnitPrice()%>원</h4>
23 <p><a href="#" > 상품 주문 &raquo;</a>
24 <a href="./products.jsp"> 상품 목록 &raquo;</a>
25 <hr>
26 <jsp:include page="footer.jsp" />
27 </body>
28 </html>
```

시작 페이지의 접속 시각 자동 갱신하기

```
welcome.jsp
10+ <%!String greeting = "웹 쇼핑몰에 오신 것을 환영합니다";
12+ <div class="jumbotron">
19- <div class="container">
20-     <div class="text-center">
21-         <h3>
22             <%=tagline%>
23         </h3>
24-         <%
25             response.setHeader("Refresh", 5);
26             Date day = new Date();
27         %>
```


상품 상세 정보 버튼 만들기(부트스트랩 적용 후)

▶ 상품 목록에 버튼 추가하기

```
22<  <div class="container">
23<    <div class="row" align="center">
24<      <%
25<        for (int i = 0; i < listOfProducts.size(); i++) {
26<          Product product = listOfProducts.get(i);
27<      <%
28<        <div class="col-md-4">
29<          <h3><%=product.getPname()%></h3>
30<          <p><%=product.getDescription()%>
31<          <p><%=product.getUnitPrice()%>원
32<          <p><a href="./product.jsp?id=<%=product.getProductId()%>"
33<            class="btn btn-secondary" role="button">상세정보 &raquo;</a>
34<        </div>
35<      <%
36<    }
37<  <%
38< </div>
```

상품 상세 정보 페이지 만들기

product.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="dto.Product"%>
3 <jsp:useBean id="productDAO" class="dao.ProductRepository"
4     scope="session" />
5 <html>
6 <head>
7 <link rel="stylesheet"
8     href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
9 <title>상품 상세 정보</title>
10 </head>
11 <body>
12 <jsp:include page="menu.jsp" />
13 <div class="jumbotron">
14 <div class="container">
15 <h1 class="display-3">상품정보</h1>
16 </div>
17 </div>
18 <%
19     String id = request.getParameter("id");
20     Product product = productDAO.getProductById(id);
21 %>
```

상품 상세 정보 페이지 만들기

```
22<div class="container">
23  <div class="row" align="center">
24    <div class="col-md-6">
25      <h3><%=product.getPname()%></h3>
26      <p><%=product.getDescription()%>
27      <p><%=product.getUnitPrice()%>원
28      <p><b>상품 코드:</b> <span class="badge badge-danger">
29        <%=product.getProductid() %></span>
30      <p><b>제조사</b> : <%=product.getManufacturer()%>
31      <p><b>분류</b> : <%=product.getCategory()%>
32      <p><b>재고 수</b> : <%=product.getUnitsInStock()%>
33      <h4><%=product.getUnitPrice()%>원</h4>
34      <p><a href="#" class="btn btn-info"> 상품 주문 &raquo;</a>
35      <a href="./products.jsp" class="btn btn-secondary"> 상품 목록 &raquo;</a>
36    </div>
37  </div>
38  <hr>
39</div>
40<jsp:include page="footer.jsp"/>
41</body>
42</html>
```