

8장 서블릿끼리 연동

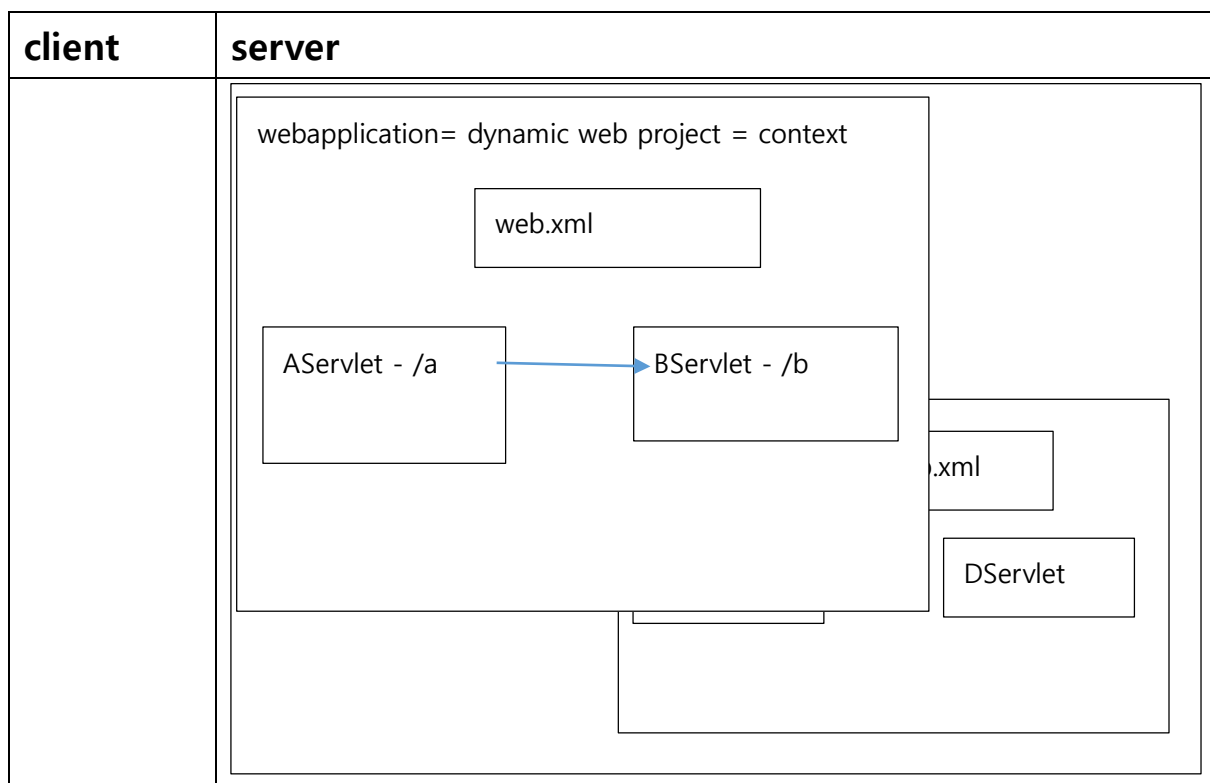
html

css

js

jquery

servletestest



HttpServlet	기본 api
HttpServletRequest	
HttpServletResponse	
RequestDispatcher	확장 api- 서블릿끼리 연동 tomcat web.xml – 서블릿 연동
ServletContext	
ServletConfig	
HttpSession	

Cookie	
--------	--

<서블릿 연동>

RequestDispatcher rd <hr/> rd.forward(request, response)	요청 - 처리 - 응답 요청공유(forward이동 서블릿까지) request.setAttribute("", 객체) request.getAttribute("") request.removeAttribute("")
ServletContext context =====	context=web application=dynamic web project (html, java, jsp) 컨텍스트공유=> 동일 dynamic web project존재 서블릿 공유 context.setAttribute("", 객체) context.getAttribute("") context.removeAttribute("")
HttpSession session	세션공유(동일 브라우저 실행 서블릿끼리) session.setAttribute("", 객체) session.getAttribute("") session.removeAttribute("")

요청-응답 종료	HttpServletRequest setAttribute("", 객체) getAttribute("") removeAttribute("")
브라우저 종료	HttpSession setAttribute("", 객체) getAttribute("") removeAttribute("")
톰캣서버 종료	ServletContext setAttribute("", 객체) getAttribute("") removeAttribute("")

쿠키	세션
클라이언트측 브라우저 크기 한계. 4kb 1개 서버당 1개. 문자열값 브라우저 종료시-브라우저쿠키 /종료 이후-파일쿠키(탐색기 폴더 파일형태 존재) c1.setMaxAge(60*60*24*7); 보안 취약 현재 서버 xxx째방문입니다	서버측 자바객체 전달 활용 1개 클라이언트당 1개 브라우저 종료시 / 종료 이전 보안 유리 로그인아이디, 장바구니 ArrayList<ProductDTO>

11장 jsp 정의 구성요소

웹클라이언트	웹서버
html – 화면구성 css –크기 색상 모양 js jquery – 동작 (화면 변경)	servlet jsp

servlet	jsp
자바 언어 구현 + html태그 응답	자바 언어 구현 + html태그 응답
자바 클래스 상속 메소드 오버라이딩 doGet / doPost / service 자바문장 내부에 html;태그 포함 String 타입 저장	html파일 유사형식 클래스 , 상속 오버라이딩 구현 필요없다(내부적 클래스 , 상속 오버라이딩 자동 변경) html파일 내부 태그 양식으로 자 바 문장 포함 형식

- jsp 실행

1>JSP 소스를 자바 소스로 변경한다.

C:\Wkdigital\workspace_web\metadata\plugins\org.eclipse.wst.s
erver.core\tmp0\work\Catalina\localhost\다이나믹웹프로젝트명
\org\apache\jsp

1-1. HTML 태그들은 out.write("") 출력

1-2. 클래스 상속 _jspService 메소드 구현.

1-3. <% %> 자바 문장 실행

2>컴파일한다

3>실행 결과 출력한다

html 태그	html태그 그대로
jsp 태그	<% 스크립트릿 태그 %> <%@ 지시자 page include %> <%! 멤버변수나 메소드 선언 %> <%=출력, 속성값 표현 %> <%-- 주석 --%>
	<jsp:xxxx />
	\${el언어}
	<c:xxx />

<%@ page

import="" contentType="text/html; charset=utf-8"

%> page 지시자태그 -directive tag

==> jsp -> 자바소스 변경 시점에서 tomcat 지시 사항 정의

---> jsp 파일 첫번째 등장 태그

page ==> 자바 this 키워드 유사역할

<%@ include %>

==> 다른 jsp 합쳐서 실행 정의

board.jsp	member.jsp	shop.jsp
화면 하단 a.png파일 이미지	화면 하단 a.png파일 이미지	화면 하단 a.png파일 이미지

3개 jsp 파일에 공통 기능 필요- 공통 기능 구현-코드반복

==>

3개 jsp 파일에

공통 기능 jsp 파일- a.jsp

board.jsp <%@ include file="a.jsp" %>	member.jsp <%@ include file="a.jsp" %>	shop.jsp <%@ include file="a.jsp" %>
share.jsp - 재사용		

board.jsp + share.jsp--> *.java 1개 생성

<%@ taglib %>

==> jsp 추가 태그 사용 알려주는 정의/

12장

<%@ page import="" %>

<% 자바 문장 ; %>

==> 스크립트릿 태그

<%! 멤버변수 + 메소드 추가 %>

==> 선언문 태그

<%! String var = "첫번째"; %>

<%

String var2 = "두번째";

S.o.p(var2);

%>

first.jsp --> first_jsp.java 자바소스 변경

= _jspService() 내부 지역코드 생성

<%

String var2 = "두번째"; ==> _jspService() 지역변수 간주

S.o.p(var2);

%>

<%! String var = "첫번째"; %> ==> jsp 변경된 자바소스의 모든
메소드 사용 멤버변수 간주

†

<%-- jsp 주석(jsp 를 자바 소스 변경 제외) --%>

<!-- html 주석 (jsp 르, 자바소스로 변경 컴파일 포함. 실행 포함., 브라우저 결과 포함 응답. 브라우저 실행 제외) -->

<%

자바 언어 주석 (jsp를 자바 소스로 변경시 포함, 자바 컴파일 제외)

//자바문장;

/* */

%>

<%-- <%@ include ... %> --%>

<!-- <%@ include ... %> -->==> error

-내장객체

jsp – 응답 html 태그 작성 쉽다.

처리결과 자바 변수 형태 제공 문장 사용 태그

<% 개발자 자주 사용 객체 – 미리 제공 - 내장객체 %>

request	요청	doGet doPost service ==> HttpServletRequest,
response	응답	HttpServletResponse,
out	응답	PrintWriter o = response.getWriter();

	<pre><% out.println(""); %></pre> <pre><%= "" %></pre>	
session	jsp-서블릿 데이터공 유	HttpSession session = request.getSession()
application	jsp-서블릿 데이터공 유	
config	jsp-서블릿 데이터공 유	
exception	<pre><%@ page isErrorPage=true %></pre> jsp파일만 내장 객체 사용	

-442p

SessionTestServlet	session.jsp	
id=jsp ==> 세션 저장	세션 저장값 id	브라우저

a.jsp <pre><%@ page errorPage="b.jsp"</pre> 10 / 요청파라미터값 --> b.jsp 예외처리 떠넘김 c.jsp <pre><%@ page errorPage="b.jsp"</pre>	b.jsp(예외처리전담jsp) <pre><%@ page isErrorPage=true</pre> 예외처리 exception 내장객체 사용가능
---	---

--	--

12장

<% %>

내장객체

13장 자바 코드를 없애는 액션 태그

-jsp 파일은 html태그 + <% 자바문장 %> 구분된다.

- BoardDAO dao = new BoardDAO();

==> <jsp:useBean id="dao" class="BoardDAO" />

BoardDAO dao = new BoardDAO();

session.setAttribute("dao", dao);

==> <jsp:useBean id="dao" class="BoardDAO" scope="session" />

<jsp:forward />

<jsp:forward > contents </jsp:forward>

<jsp:useBean

<jsp:setProperty

<jsp:getProperty

<jsp:include />

<%@ include file="share.jsp" %>

board.jsp	member.jsp	shop.jsp
share.jsp drinks.gif 출력		

```

<jsp:include page="share.jsp" >
    <jsp:param name="img" value="s.png" />
</jsp:include>

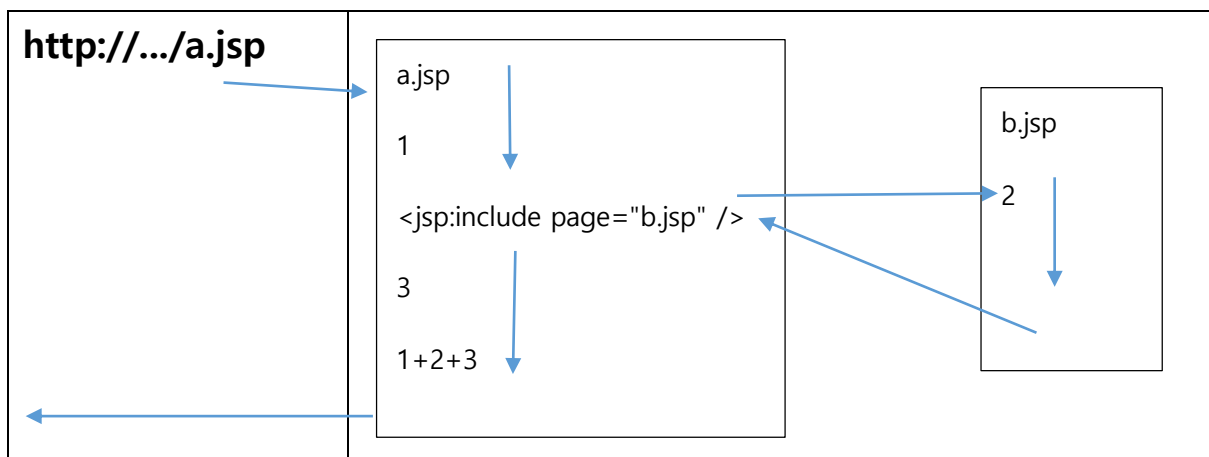
```

board.jsp b.png	member.jsp m.png	shop.jsp s.png
share.jsp ? 출력		

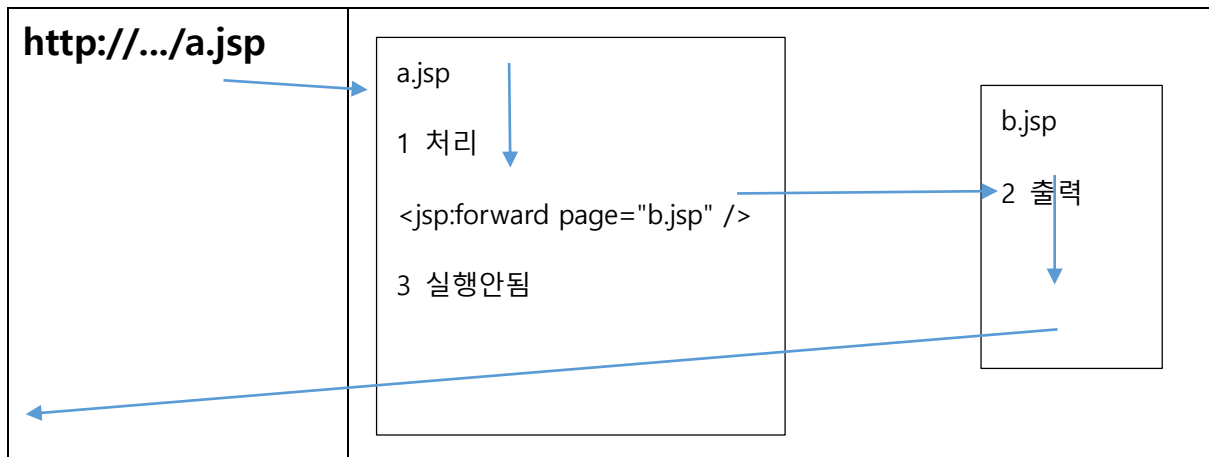
동적 이미지 변경되어지면서 include

<jsp:include 액션=동적	<%@ include 디렉티브=정적
재사용 화면 구성시 레이아웃 설계	재사용 화면 구성시 레이아웃 설계
2개 jsp=> 2 개 자바 소스	2개 jsp => 1 개 자바 소스
<jsp:param > 파라미터 전달 값 >	변수 공유 가능 <% String name %>

include



forward



-include 파라미터 전달x

```
<jsp:include page="" />
```

-include 파라미터 전달o

```
<jsp:include page="" >
```

```
<jsp:param name="" value="" />
```

```
</jsp:include>
```

-include 자바객체 전달o

```
request.setAttribute("", 객체);
```

```
<jsp:include page="" />
```

- 파라미터 없이 이동

```
<jsp:forward page="" />
```

- 파라미터 갖고 이동

```
<jsp:forward page="" >
```

```
<jsp:param name="" value="" />
```

```
</jsp:forward>
```

- 이동시 자바객체 전달o

```
request.setAttribute("", 객체);
```

```
<jsp:forward page="" />
```

서블릿 RequestDispatcher d = request. getRequest...("jsp파일명"); request.setAttribute("", 객체); d.forward(request, response) d.include(request, response)	jsp request.setAttribute("", 객체); <jsp:forward page="서블릿 url"/> <jsp:include page="서블릿 url"/>
--	--

-

BoardServlet.java (/boardstart?page=1) BoardDAO 객체 생성 getBoardList(, 호출하면 ArrayList<BoardDTO> getBoardCount() 메소드 호출한 결과를 전달 boardstart.jsp로 forward	boardstart.jsp 리스트 테이블태그 1 2 3 4	BoardDAO getBoardList getBoardList(, getBoardCount()
		BoardDTO

서블릿-dao, dto -서블릿
jsp-dao, dto -jsp

서블릿과 jsp 연동

