

web application = web context : 웹서비스 위해 사용 여러가지 파일 모음

src:패키지명.*.java==> 서블릿

webcontent:*.jsp

webcontent:*.html *.css *.js *.jpg

webcontent\WEB-INF:web.xml(웹서비스 환경설정파일)

웹클라이언트 html css js jquery

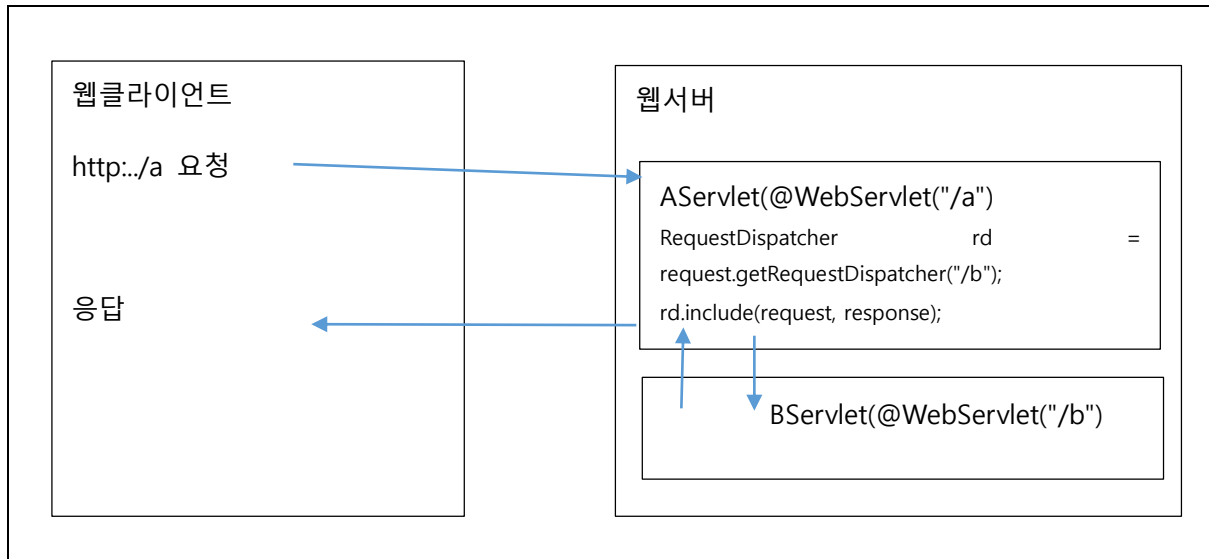
웹서버 (tomcat+jdk) servlet jsp spring

servlet

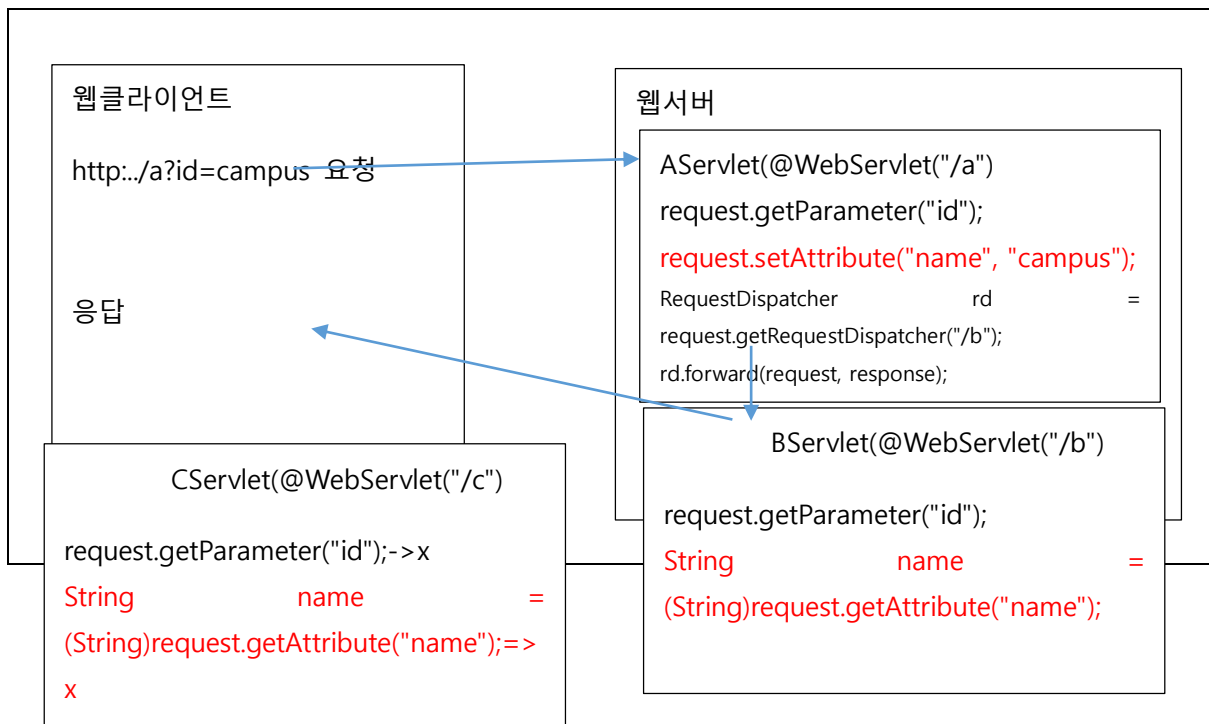
작성 4.0 기준 tomcat 9+jdk 8	@WebServlet("/a")--> 3.0 이상 추가 class A extends HttpServlet doGet / doPost 오버라이딩 ==> web.xml 파일 대신 설정 1.0 <servlet> <servlet-name> (임시)서블릿이름 </ <servlet-class> 패키지명.서블릿실제클래스명 </ </servlet> <servlet-mapping> <servlet-name> (임시)서블릿이름 </ <url-pattern> /a </ </servlet-mapping>
실행	http://ip:port/컨텍스트명/a
클라이언트로부터 입력받기	요청 doGet(HttpServletRequest request, HttpServletResponse response) String id = request.getParameter("id"); ==> 정수라면 Integer.parseInt(id); String [] <==request.getParameterValues("fruit");

db 연동하기	<p>처리</p> <p>로그인 회원가입 게시물조회</p> <p>--> db 조회 / 저장</p> <pre> class A extends HttpServlet{ doGet(.....){ //요청 jdbc //응답 } </pre>
자바 클래스 연동하기	<p>처리</p> <p>XXXDAO(각 기능 1개(SQL) 메소드) / xxxDTO :xxxVO (레코드 1개 표현 객체)</p> <pre> XXXDAO dao = new XXXDAO(); ?=dao.메소드명(?); </pre>
다른 서블릿과 연동하기	<p>처리-응답</p> <pre> RequestDispatcher rd = request.getRequestDispatcher("/b"); rd.include(request, response); rd.forward(request, response); ==> tomcat 요청 b=new BServlet(); b.init(); b.doGet() </pre>

- include



-forward



api 이름

상속	javax.servlet.http.HttpServlet
매개변수 선언	javax.servlet.http.HttpServletRequest javax.servlet.http.HttpServletResponse
다른 서블릿 포함/이동 포함/이동 서블릿과 데이터 공유	javax.servlet.RequestDispatcher request.setAttribute("", ""); request.getAttribute(""); request.removeAttribute("");
다른 서블릿과 데이터 공유	javax.servlet.http.Cookie
다른 서블릿과 데이터 공유	javax.servlet.http.HttpSession session.setAttribute("", ""); session.getAttribute(""); session.removeAttribute("");

--	--

세션과 쿠키

1> http 프로토콜 url 요청한다

2> get/post 파라미터 전송 방식이 있다

3> 첫번째 요청(id=test) -연결생성-처리-응답-연결해제 (요청/응답객체삭제)

==> doGet() / doPost() 완료 - 삭제

두번째 요청 - id=test 정보 서버에서 남아있지 않다 - 생성-처리-응답-연결해제

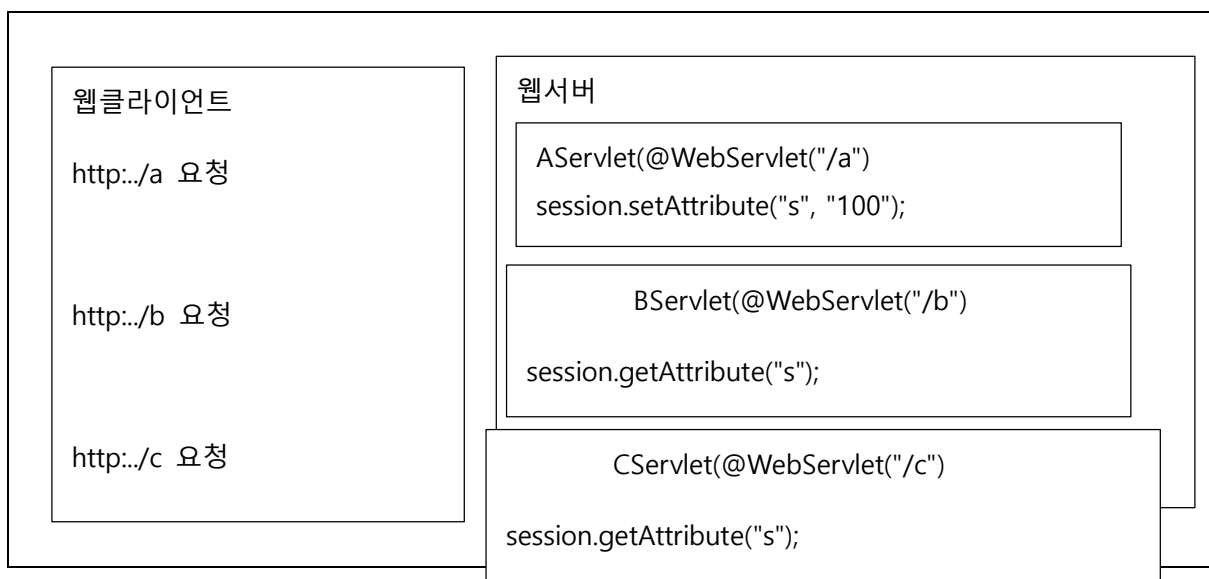
4> http 프로토콜 서버는 모든 클라이언트의 정보를 서버에 남기지 않는다

==> 1개 서버가 다수개 클라이언트 처리=메모리한계

5> id=test 로그인하면 메일 게시판 장바구니 이전방문날짜 방문횟수

6> <input type=hidden name="id" value="test">

==>?id=test



- 서블릿 데이터 공유

쿠키	세션
브라우저 종료시 혹은 종료 이후 지정 시각까지	브라우저 종료시
클라이언트 컴퓨터측 저장 기술	서버 컴퓨터측 저장 기술
문자열 저장 가능 (인코딩설정 영문 그대로 숫자 그대로 한글 인코딩)	자바객체 저장 가능
보안 누출 (영문 암호화)	보안 유리

1. 클라이언트 요청한다 4. 쿠키 내용이 브라우저 전송된다.	2. 서버에서 쿠키 객체로 생성한다 <code>Cookie c = new Cookie("쿠키명", "");</code> <code>c.setMaxAge(60*60*24);</code> 3. 클라이언트 응답한다 <code>response.addCookie(c);</code>
5. 클라이언트 요청한다 (동일 서버로부터 받은 쿠키도 같이 전송) 4	6. 클라이언트로부터 쿠키를 읽어온다 <code>Cookies[] coo= request.getCookies();</code> <code>coo[0].getName();</code> <code>coo[0].getValue();</code> 2,3

<p>1. 클라이언트가 요청한다</p> <p>요청</p>	<p>2. 처리-세션 데이터 저장 -공유</p> <p>HttpSession session = request.getSession();</p> <p>false – 최초 연결</p> <p>3. session.setAttribute("세션속성명", 값(Object));</p>
<p>4. 클라이언트가 요청한다</p> <p>7. 응답</p>	<p>5. HttpSession session = request.getSession();</p> <p>true – 이전 연결 중</p> <p>6. Object o = session.getAttribute("세션속성명");</p> <p>=>원래 타입 형변환 필요시</p>
<p>8. 클라이언트 요청한다(로그아웃 경우처럼 서버 남겨놓은 정보 삭제)</p>	<p>9. HttpSession session = request.getSession();</p> <p>true – 이전 연결 중</p> <p>10. session.removeAttribute("세션속성명");</p>

<p>homepage.html</p> <p>로그인</p> <p>로그아웃</p> <p>내정보조회</p>	<div> <p>로그인 링크 -</p> <p>logindb.html 복사하여 loginsession.html 이름 변경(form action="..."</p> <p>loginsession.html 이동</p> <p>id, password 2개 입력하여 get 방식으로 loginsession url 매핑 서브릿 전송</p> <p>session.LoginSessionServlet.java</p> <p>@WebServlet("/loginsession")</p> <p>doGet ; id 를 세션 객체의 속성 저장 ("sessionid", 폼입력id)</p> </div> <div> <p>로그아웃 링크 -</p> <p>session.LogoutServlet.java("/logout")</p> <p>doGet :</p> <p>로그인한 적 있을 때에만 세션 객체 저장 id값 삭제</p> <p>(로그인한 적 있을 때에만 로그아웃)</p> <p>로그인한 적 없을 때 로그인부터 하세요 출력</p> </div> <div> <p>내정보 조회 링크 -</p> <p>session.MyInformServlet.java("/myinform")</p> <p>doGet</p> <p>세션 객체의 속성 회원님 환영합니다.</p> <p>메일 보러 가기</p> <p>게시판 글쓰러 가기</p> </div>
--	--

servlet / jsp

자바 웹서버 - 서블릿 구조 제공 + jsp 구조 제공
(4.0 2.3)

서블릿 *.java(src..)	jsp (webcontent\WxxxW*.jsp)
<pre>class A extends HttpServlet doGet(){ 응답 - 자바 언어 내부 html 언어 포 함 형식 out.println("<h1>..</h1>"); } doPost</pre>	<pre><% 서블릿 api + db + DAO, VO 객체 생성 자바 문장 ; %> <HTML> <BODY> HTML 양식 언어 내부 자바 문장 포 함 JSP 태그와 HTML 태그 혼용</pre>
서블릿 api 사용 처리	서블릿 api + html 언어 독립적

jsp - java server page

html + java

작성 - 실행

- jsp 내부 tomcat server 서블릿 변환 -(*.java) - 컴파일 (*.class)- 객체생성 -
init - doGet...

===>

1> servlet api + jsp api + 내장객체들

2> <% %>->jsp 기본 태그

<jsp:xxxx ->jsp 액션 태그

<c:xxxx -> custom 태그

\${x} -> el

3> 1개웹페이지 = servlet + jsp 연동

4> spring

html	html css js jquery
jsp 기본	<% _jspService() 메소드 내부 지역변수 선언 포함 자바 문장 ; %> 표현문태그 - <%= 변수명(값) %> 선언문태그 - <%! 메소드 정의, 필드변수 정의 %> jsp주석태그 - <%-- --%> 지시문태그 (디렉티브 태그) - <%@ %> <%@ page text/html; charset=utf-8 <%@ include ==> 13장 <jsp:include>비교 <%@ taglib ==> 14장
jsp action	<jsp:xxxx />
jsp custom	<c:xxx />
el	\${ i } <script> \$("#i") </script>
내장객체	jsp는 서블릿보다 간결 응답 , servlet api request response exception session session.xxxx