

javax.servlet.http.HttpServlet	상속 doGet(HttpServletRequest request, HttpServletResponse res) doPost(HttpServletRequest req, HttpServletResponse res)
javax.servlet.http.HttpServletRequest	요청정보객체 request.getParameter("") request.getParameterValues("") request.getRemoteAddr()->서버 기록(로깅) logging request.setCharacterEncoding("utf-8")
javax.servlet.http.HttpServletResponse	응답정보객체 response.setContentType("text/html;charset=utf-8"); PrintWriter o = response.getWriter() o.println("<h1>텍스트...") response.addHeader() response.addCookie()
javax.sql.DataSource	웹 "미리 db연결생성-일정갯수 정해놓자-재사용" 웹서버가 제공

javax.servlet패키지

javax.servlet.http패키지

8장 확장 api

javax.servlet.RequestDispatcher	동일 web application 내부 다른 파일 연결 = context = dynamic web project srcW*.java webcontentW*.html *.css *.js *.jsp *.mp3...
javax.servlet.ServletContext	동일 web application 내부 다른 파일과 데이

	터 공유
--	------

9장

javax.servlet.http.Cookie	동일 web browser 내부 다른 파일과 데이터 공유
javax.servlet.http.HttpSession	동일 web browser 내부 다른 파일과 데이터 공유

=====

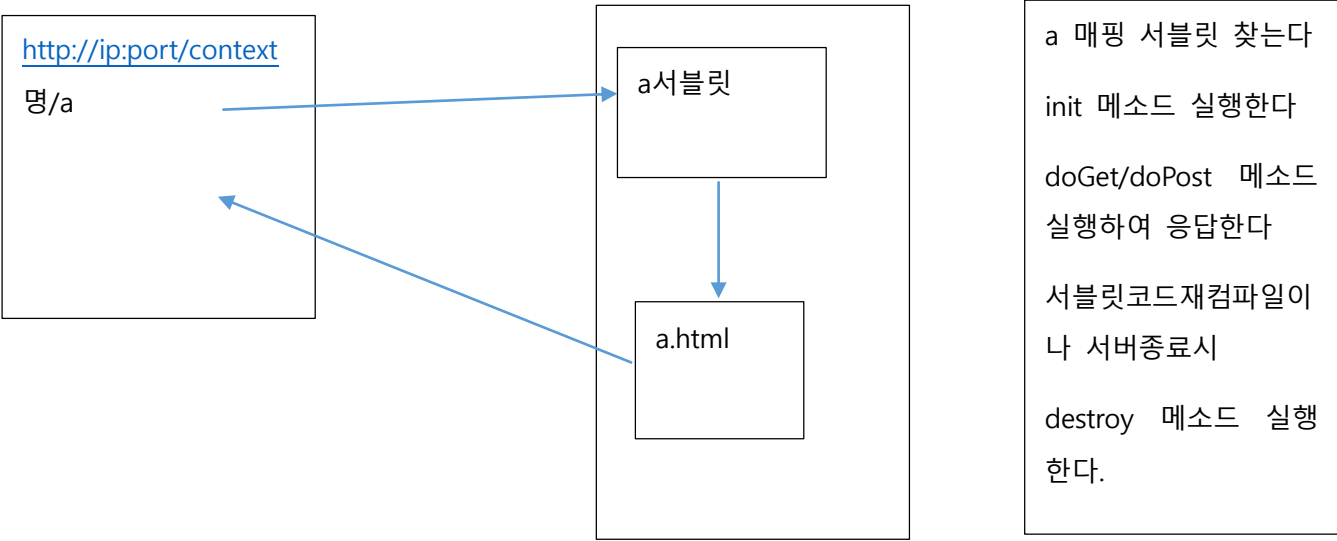
8장

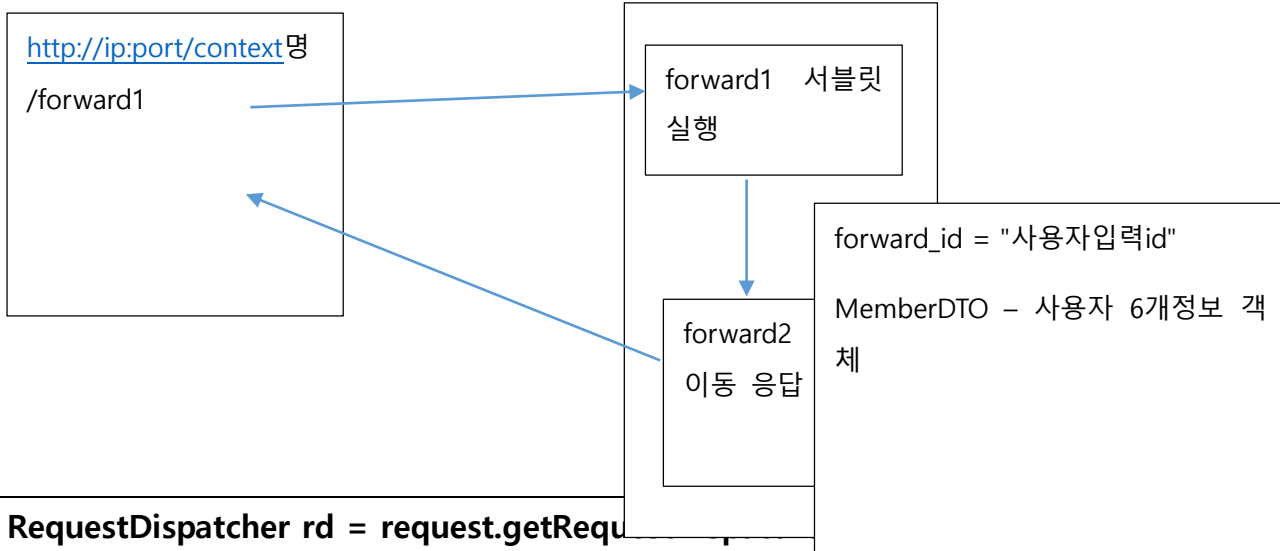
forward – 이동한다

1> 서블릿 자바코드 그대로 사용 db 연결 처리 구현

2> html 브라우저 출력 구현

javax.servlet.RequestDispatcher





```

RequestDispatcher rd = request.getRequestDispatcher("/forward2");
RequestDispatcher rd = request.getRequestDispatcher("/forward2.html");
RequestDispatcher rd = request.getRequestDispatcher("forward2.html");
RequestDispatcher rd = request.getRequestDispatcher("forward2");
  
```

request.getParameter("id") – 클라이언트 정보 id
 request.getAttribute("id") – 서블릿끼리. 이동직전 request.setAttribute("id", "xxxx");
 rd.forward(request, response);

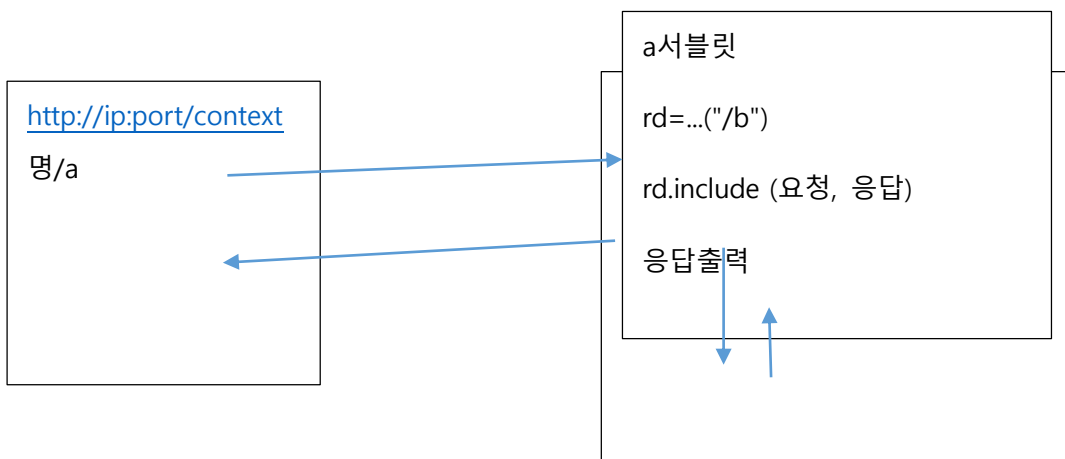
request.getRequestURI(); 현재실행중인 서블릿 uri값

url => 네트워크 호출 파일 구분 방법

<http://ip:port /컨텍스트명/파일명>

uri ==> 서버 컴퓨터 내부 호출 파일 구분 방법

</컨텍스트명/파일명>



b서블릿

응답출력

a	b
<pre>doGet(,,,request, ..response) RequestDispatcher rd = request.getRequestDispatcher("/b"); request.setAttribute("id", "java") rd.forward(request, response);</pre>	<p>출력 내용 구현</p> <pre>String id = (String)request.getAttribute("id")</pre>

a	b
<pre>doGet(,,,request, ..response) RequestDispatcher rd = request.getRequestDispatcher("/b"); 출력1 request.setAttribute("속성명", "전달값") rd.include(request, response); 출력3</pre>	<p>출력2 내용 구현</p> <pre>request.getAttribute("속성명")</pre>

-객체전달

a	b
<pre>doGet(,,,request, ..response) RequestDispatcher rd = request.getRequestDispatcher("/b"); request.setAttribute("dto", new Memb erDTO()) rd.forward(request, response);</pre>	<p>출력 내용 구현</p> <pre>MemberDTO o = (MemberDTO)request.getAttribute("dto"); request.getRequestURI();</pre>

login.AllUserServlet ==> dispatch.AllUserForwardServlet 복사.

@WebServlet ("/alluserforward") 변경

http://ip:port/servlettest
/alluserforward

alluserforward

1> MemberDAO

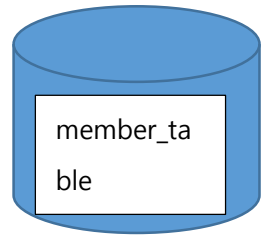
SELECTMember()

호출 결과

ArrayList<MemberDTO> 리턴

2> alluserlist 서블릿 정의.
forward

3> 1번에서 리턴받은
ArrayList<MemberDTO> 출력



-동일 서버 동일 컨텍스트 내부 파일 공유

데이터(문자열, 정수, 실수, 객체(ARRAYLIST, MEMBERDTO..)) 공유 기간

요청 처리 응답 동안(요청 처리 동안) 8장	include /forward 파일 실행 끝날때까지 RequestDispatcher request.setAttribute("이름", 객체); (클래스명)request.getAttribute("이름") include(request, response) / forward(request, response)
브라우저 지속 동안 9장	Cookie , HttpSession
동일 컨텍스트 모든 서블릿 실행 동안 (서버 지속 동안 / 브라우저 닫고 재실행해도 정보 유지) 8장	ServletContext

```

ServletContext context = getServletContext(); //현재서블릿 컨텍스트정보 객체
context.setAttribute("", 객체); //현재 컨텍스트 내의 모든 파일과 공유. 서버 종료 기간
context.getAttribute("");
context.removeAttribute();

```

a

MemberDTO 객체 생성

```

context.setAttribute("m",
new MemberDTO(...));

```

b

```

context.getAttribute("m");

```

c

MemberDTO 객체 공유

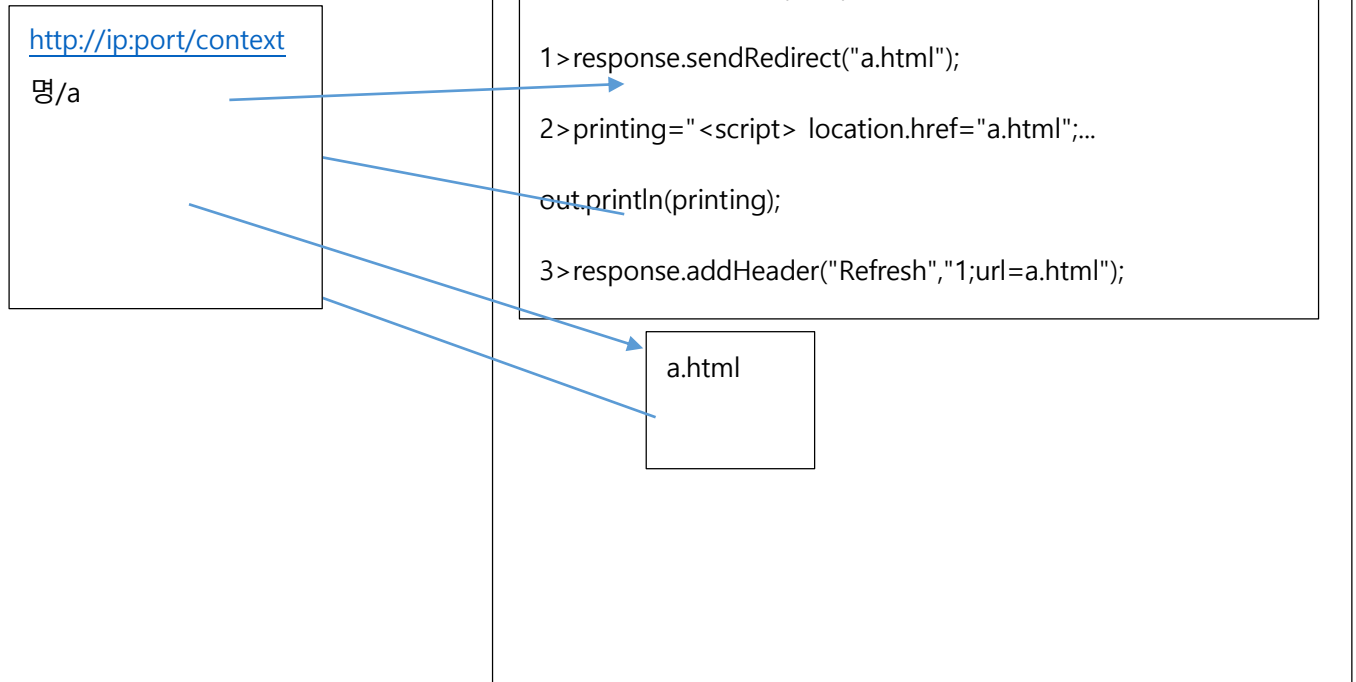
```

context.getAttribute("m");

필요없다 삭제
context.removeAttribute("m");

```

요청처리	서버종료이전	???
request.setAttribute("", 객체) request.getAttribute("") request.removeAttribute("") forward/include(request,.....)	ServletContext context = getServletContext(); context.setAttribute("", 객체) context.getAttribute("") context.removeAttribute("")	



9장 쿠키와 세션 알아보기

브라우저 내부(닫기전까지) 실행 파일들 데이터 공유
같은 서버 같은 컨텍스트 내부 파일들만 대상

쿠키	세션
Cookie 클래스 클라이언트측 텍스트파일 저장 공유 브라우저 제공 툴 정보 확인 가능 브라우저 닫을때까지 정보 공유 기본기간 + 추가선택적 (1일 30일 365일) 당신은 xxxx번째 방문하셨습니다 자바객체 생성 공유 불가능	HttpSession 서버측 메모리 저장 공유 보안 민감 데이터 적절 브라우저 닫을때까지 정보 공유 기본기간 + 브라우저 닫지 않고 30분 동안 액션 없으면 세션 자동 삭제 로그인정보 자바객체 생성 공유 가능

1.http://ip:port/context명/login
6. 결과 받는다.
(자동 파일 생성- id:spring)
7. http://ip:port/context명/writing
(자동 쿠키 서버로 전송)
10. http://ip:port/context명/cart
(자동 쿠키 서버로 전송)

2. a 서블릿 메소드 실행한다
3. 저장 필요 변수값 Cookie 객체로 생성
Cookie c = new Cookie("id", "**spring**");
4. 쿠키 필요 추가 사항 설정
c.setMaxAge(60*60*24);
5. 클라이언트로 전송
response.addCookie(c);
8.b 서블릿 메소드 실행한다.
Cookie [] ar = request.getCookies();
9. ar[0].getName()
ar[0].getValue()

- cookie

- 1> 서버에서 클라이언트로 데이터 전송하고
- 2> 내부 저장 영문자 숫자 저장
- 3> 공백 특수문자 일부 오류
- 4> 한글 저장

bookcookie?title=spring완성

지금까지 선택하신 도서는 spring완성을 선택하셨습니다.

bookcookie?title=react완성

지금까지 선택하신 도서는 spring완성, react 완성을 선택하셨습니다.