

4장 (http 개념용어) 웹어플리케이션 이름
 = (이클립스용어) dynamic web project이름
 =(tomcat서버) context이름

<http://ip:port/context이름/servletmapping>

dynamic web project- test 다양한 파일들 포함 1개 프로젝트 생성 위해 필요한 파일들 = web application – test = context - test	java resources 폴더 jdk라이브러리폴더 src₩*.java==.서블릿클래스
	WEBCONTENT 폴더 ₩ a.html http://ip:port/test/a.html WEBCONTENT₩WEB-INF폴더₩a.html http://ip:port/test/WEB-INF/a.html ==> FILE NOT FOUND 404오류 WEBCONTENT₩WEB-INF폴더 WEBCONTENT₩ *.HTML *.css *.js *.jpg *.mp3 *.mp4 *.jsp

서블릿 매핑

1> 3 이후 @WebServlet 사용하기—자바소스 내부 직접 작성—단순/컨텍스트 전체 서블릿매핑이름 모두 파악 어려운 구조

2> 3 이전 -- web.xml 사용하기-표현태그 많다/컨텍스트 모든 서블릿 매핑이름 한병네 파악 쉬운 구조

1> 서블릿 클래스 정의

file-new-servlet- 패키지명, 클래스명 입력 – url mapping ('/FirstServlet')-종료

```
package test
@WebServlet("/FirstServlet")
class FirstServlet extends HttpServlet { ==> A 서블릿 정의=server side applet(자바클래스)
```

```
doGet(){ 실행 결과 요청 브라우저 전송 }
```

자동 컴파일

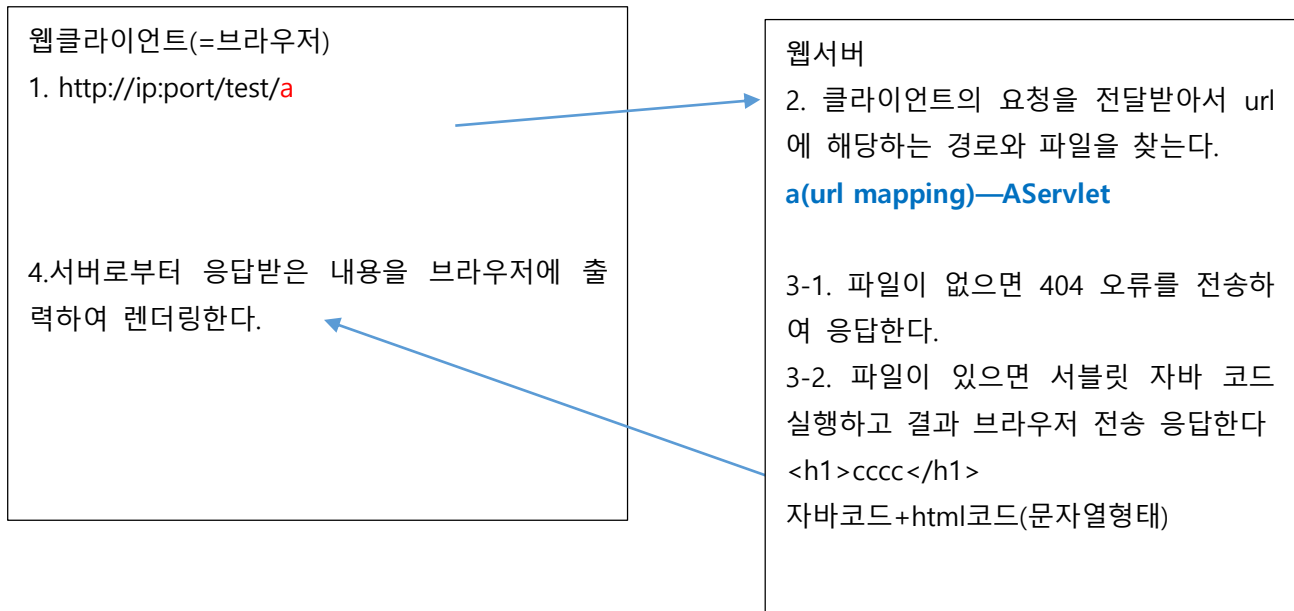
context명/**FirstServlet**

2> 실행

run – run on server – 브라우저 자동 오픈 주소 입력

<http://ip:port/servlettest/FirstServlet> 요청시

Servlet 3 이상 1. @WebServlet 추가 2. web.xml 파일 활용	url mapping시 @WebServlet("/second") class A...{
Servlet 2.5 web.xml 파일 활용	url mapping시 <servlet> <servlet-name>ss</servlet-name> <servlet-class>SecondServlet</servlet-class> </servlet> <servlet-mapping> <servlet-name>ss</servlet-name> <url-pattern>/second</url-pattern> </servlet-mapping>



- 서블릿 실행 흐름

클라이언트	서버
1. 서블릿 요청(/a)	2. 서블릿컴파일클래스 찾아서 객체 생성. 메모리 로딩
6. 브라우저 출력	3. init 메소드 호출
7. 서블릿 요청(/a)	4. 8. doGet 메소드 호출- 요청/응답
10. 브라우저 출력	5. 9. 응답 브라우저 전송
	11.서버종료 / 서블릿수정
	메모리정리=서블릿삭제
	destroy() 메소드 호출

4,5장

6장 요청과 응답

1>클라이언트 **요청**받는다

2> **요청 분석**(로그인, 회원가입, 게시물1개, 다른페이지이동,...) **처리**(자바 로직)

db 연결 – 회원정보테이블 id, pw 존재 판단한다

3>"정상로그인" , "비정상로그인" 클라이언트에게 **응답**한다

<pre>@WebServlet("/a") class A extends HttpServlet{ doGet(HttpServletRequest request , HttpServletResponse response) { request.xxxx() / 자바모든로직 / response.xxxx() } }</pre>
--

<p>login.html</p> <pre><form action="http://localhost:9090/servlettest/login" <input...type=text name="i" id="d" > <input .. <input type=submit value="로그인"</pre>	<p>login-->LoginServlet.java</p> <pre>doGet(..HttpServletRequest request , HttpServletResponse response){ request.getParameter("i"); ...</pre>
--	---

요청	<pre>String <= request.getParameter("input 태그 name속성값"); String [] <= request.getParameterValues("input type=checkbox name속성값"); <input type=checkbox <select multiple</pre>
처리	
응답	<pre>response.setContentType("text/html;charset=utf-8"); PrintWriter out = response.getWriter(); out.println("<h1>.....");--> 브라우저출력 System.out.println("");--> 서버실행이클립스콘솔창출력</pre>

doGet / doPost

get방식요청처리 / post방식요청처리

- html 태그 속성내부 url 표현

<form action=""

<a href=""

<script src="http://xxx/react.js"

<link href=""

절대url	http://.... 외부서버의 파일 url
상대url(현재서버root 기준)	http://localhost:9090 --> '/' http://localhost:9090/servlettest/login /servlettest/login
상대url(현재파일기준)	http://localhost:9090/servlettest/a.html http://localhost:9090/servlettest/login login
동일url	http://localhost:9090/servlettest/login /servlettest/login login
서블릿 비교 (현재 컨텍스트 기준)	@WebServlet("/login") '/' ==> 현재 컨텍스트 기준 http://localhost:9090/servlettest

<http://localhost:9090/servlettest/insert?id=aaa&pw=111&name=111>==>

폼가입양식 통해서만 입력

?

id=bluejeansh

&pw=1234

&name=%EC%A1%B0%EC%84%B1%ED%9D%AC

&phone=010-1111-2222

&email=bluejeansh%40hanmail.net

&address=%EC%84%9C%EC%9A%B8 --> 서울

&hobby=movie

&hobby=music

&hobby=eating+show

&hobby=study

클라이언트 폼 서블릿 전달 방식 2개-'http' 프로토콜

get	<ol style="list-style-type: none">1. 입력 데이터를 주소 노출2. url?변수명1=값1&변수명2=값2.....3. 웹서버 url 길이 제한4. 파일 업로드 불가능5. 암호 보안 민감 데이터 전송 부적절6. url 공백 불가 -> '+' 변경7. url전송데이터 암호화 <p>공백- + 영문 숫자 그대로 전송 한글 다국어 %16진수 인코딩</p> <p>html</p> <p><form action="" method=get></p> <p>또는</p> <p><form action=""></p> <p>serlvet</p> <p>doGetMethod 오버라이딩</p>
post	<ol style="list-style-type: none">1. 입력 데이터를 주소 노출하지 않고 전송2. 따로 전송 - 변수명1=값1&변수명2=값2.....3. 데이터길이 무제한4. 파일 업로드 가능5. 암호 보안 민감 데이터 전송 적절6. url 공백 불가 -> '+' 변경7. url전송데이터 암호화 <p>공백- + 영문 숫자 그대로 전송 한글 다국어 %16진수 인코딩</p> <p>html</p> <p><form action="" method=post></p> <p>serlvet</p> <p>doPost메소드 오버라이딩</p>

-http 프로토콜 정하는 규칙 (=spec.) 요청 = 서블릿 HttpServletRequest 객체

1. 시작url /insert?한글기본자동인코딩..... http버전 요청방식-get/post
2. 요청헤더
3. 요청바디 변수명=한글값&... request.setCharacterEncoding("utf-8");

-http 전송방식 규정

<pre><form method=get> <form > 회원가입하러 가 기 주소 입력</pre>	<pre>doGet(){ request.getParameter("") }</pre>
<pre><form method=post></pre>	<pre>doPost(){ request.setCharacterEncoding("utf-8"); request.getParameter("")</pre>

-http오류

404	http:ip:port/context이름/a a 매핑 서블릿 없다 @WebServlet("/aa") class A extends HttpSevlet{}
405	get방식 호출 – 서블릿 doPost 오버라이딩 post방식 호출 – 서블릿 doGet 오버라이딩

- 계산기

calculator.html	CalcServlet.java
폼--> action="calc" 숫자1 10입력 숫자2 20입력 연산자 select option : + - * / % 연산 선택 가능 + 선택하고 submit 클릭했다	@WebServlet("/calc") 10 + 20 = 30 출력

요청 – 처리(db연동필요) – 응답

-db

1>sql 문법 + mysql db

2> servlet – 웹서버 실행 자바파일

3> jdbc 문법

-필요한 db정보

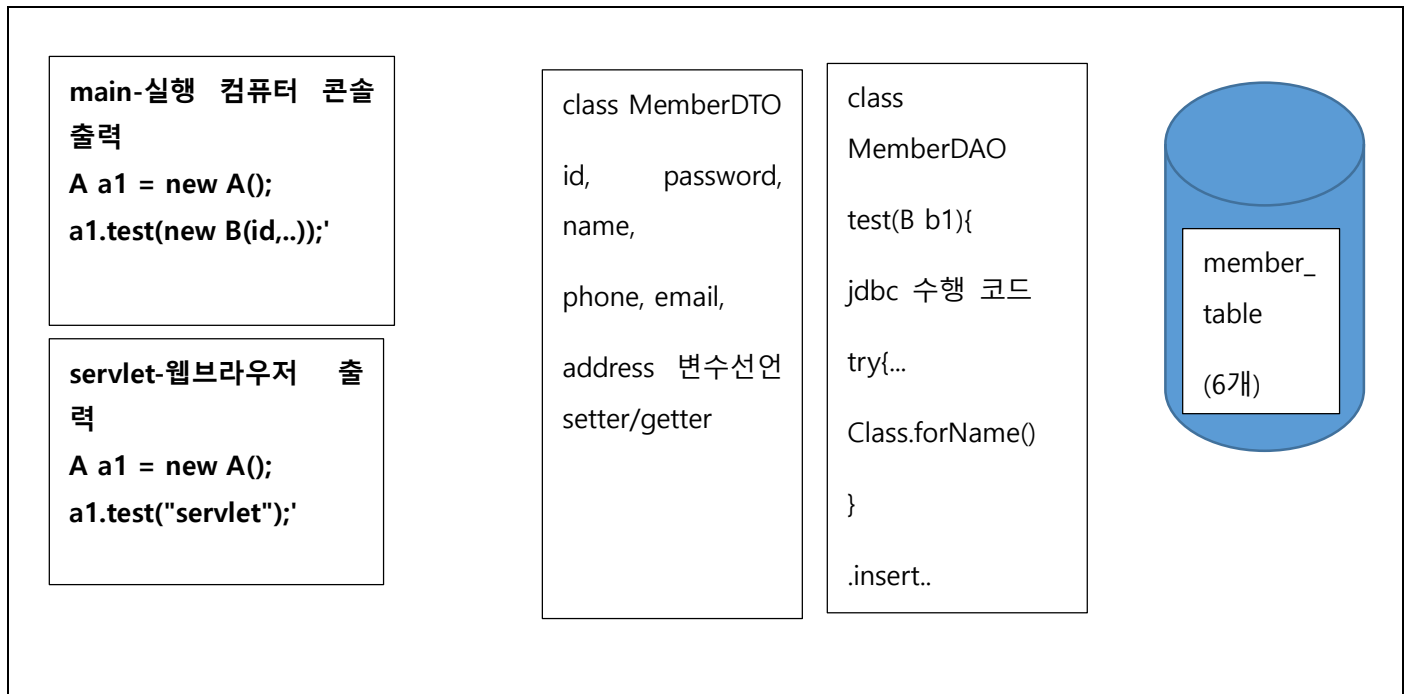
db이름 : mysql
db위치 : localhost:3306
db이름 : employeesdb
계정/암호 : emp/emp
테이블이름 : member_table

- 자바 구현 jdbc 패턴

XXXDAO 클래스 – "약속" – DATA ACCESS OBJECT = DB 접근 코드 = JDBC

XXXDTO 클래스 – "약속" – DATA TRANSFER OBJECT =

XXXVO = XXXXDO VALUE OBJECT / DATA OBJECT



실습

logindbservlet.html	LoginDBServlet	MemberDAO
<pre><form action="logindb" id pw 로그인</pre>	<pre>doGet request.getParameter(); MemberDAO dao = new MemberDAO(); int result = dao.selectMember(id, pw); if(result == 0) 정상 사용자입니다. 출력 else if(result == 1) logindbservlet.html로 링크 태그 출 력 else if(result == 2) insertservlet.html 로 링크 태그 출 력 response.setCon...()</pre>	<pre>public int selectMember(String id, String pw){ id member_table에 존재하는지 확 인 존재한다면 해당id의 db암호와 pw암호 같은지 비교 같다 return 0 같지 않다 return 1 id 존재하지 않는다면 return 2 }</pre>