SERVER

1. Port, SERVER, WAS의 개념

* 포트:

장치에서 데이터를 전송하거나 수신하는 하드웨어의 위치. HTTP는 80번 포트, HTTPS는 443번 포트를 사용.

* 웹서버:

사용자에게 웹을 제공하기 위한 서버. 웹 브라우저와 같은 클라이언트의 요청이 있는 경우 이를 받아들여 요청한 페이지를 응답. 주로 HTML, CSS, JS, 이미지파일 등 정적 컨텐츠를 리턴. Apache, Nginx

* WAS ( Web Application Server):

웹 애플리케이션과 서버 환경을 만들어 동작시키는 기능을 제공하는 소프트웨어 프레임워크. 인터넷 상에서 HTTP를 통해 사용자 컴퓨터나 장치에 애플리케이션을 수행해 주는 미들웨어(소프트웨어 엔진). 비즈니스 로직, 동적 콘텐츠를 처리하고 리턴. Tomcat, Weblogic, Jeus

* 사용자가 웹 브라우저를 통해 페이지에 접속 –> L7(url, id차단, 요청 분배) -> 웹서버(정적 파일 처리) -> WAS(톰캣. 동적 컨텐츠, 비즈니스 로직 등 처리하여 리턴)

1. 톰캣

: 웹서버에서 서블릿(servlet), JSP와 같은 자바 코드를 이해하기 위해서는 톰캣 같은 자바를 이해하는 엔진이 있어야 함(jsp를 실행하는 웹컨테이너, 서블릿컨테이너).

: WAS, 웹서버의 기능을 동시에 수행. 하나의 서버장비에 여러 개의 톰캣을 구성할 수 있으며 하나의 톰캣에 여러 개의 애플리케이션을 구성할 수 있음

1. 톰캣 다운로드, 설치, 실행, 디렉토리 구조

3-1) 설치

구글 – Tomcat 7.0.94 다운로드 – 압축풀기 – c드라이브내 JSP 폴더 내에 저장 - 환경변수 설정: 내컴퓨터 - 속성 - 고급시스템설정 – 환경변수 – 새 시스템 변수 – 변수 이름: JRE\_HOME 변수 값: C:\Program Files\Java\jre7

3-2) 실행

Cmd – cd C:\JSP\apache-tomcat-7.0.94\bin>STARTUP

웹브라우저 – localhost:8080 / <http://127.0.0.1:8080/> 접속

C:\JSP\apache-tomcat-7.0.94\bin>SHUTDOWN 페이지 접속 불가

3-3) 테스트 폴더 만들기

C:\JSP\apache-tomcat-7.0.94\webapps 내에 test폴더 만들기 – test1.jsp 파일 만든 후 저장.

웹브라우저 <http://127.0.0.1:8080/test/test1.jsp> -> 생성한 파일 확인 가능

3-4) 톰캣 접속 포트번호 변경하기

1. 기본 포트 확인: C:\JSP\apache-tomcat-7.0.94\conf 내 **server.xml** 설정파일 – 문서 중간쯤 <Connector port="**8080**" redirectPort="**8443**" connectionTimeout="**20000**" protocol="**HTTP/1.1**"/>
2. “80”으로 변경하기

Server.xml 파일 우클릭 – 편집 – Connector port="80" – 저장 – 웹 브라우저 [http://localhost:80](http://localhost/) / 127.0.0.1:80

<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

3-5) 톰캣 웹 루트 디렉토리 찾기

웹 루트 디렉토리: <http://localhost:8080/> 접속 시 찾아가는 디렉토리

경로: C:\JSP\apache-tomcat-7.0.94\webapps\ROOT

기본 설정 내용:

Server.xml에서 - <Engine name="**Catalina**" defaultHost="**localhost**">

<Host name="**localhost**" autoDeploy="**true**" unpackWARs="**true**" appBase="**webapps**">

생략내용: 엔진네임 태그 내 <Context path = “” docBase = “ROOT”/> 부분은 생략되어있음

Host의 appBase인 webapps 디렉토리의 ROOT 디렉토리를 웹 루트 디렉토리로 판단

1. 인코딩, 시작페이지 변경

4-1) 톰캣 시작페이지 변경하기

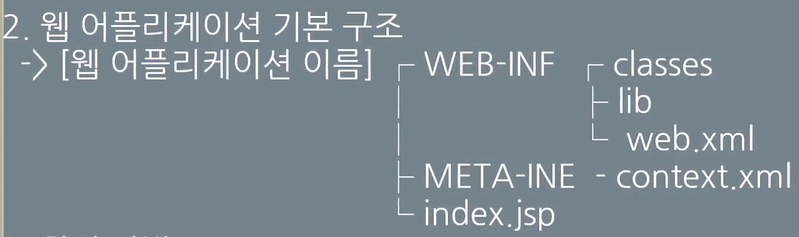
<http://localhost:8080/> 접속시 웹 루트 디렉토리의 index.jsp 파일을 불러옴 – 새로운 파일 (newindex.jsp)작성하여 루트 디렉토리에 저장 – web.xml 파일 열기 – 맨 아래 welcome-file-list 태그 내 맨 윗 줄에 **<welcome-file>newindex.jsp</welcome-file>** 추가 – 서버 재시작 후 메인 접속하면 새로 작성한 파일 내용 확인 가능

4-2) 톰캣 시작페이지 우선순위 변경하기

welcome-file-list 태그 내 순서대로 우선순위 적용. -> 순서를 바꿀 때 마다 서버 재구동해줘야함

1. 톰캣 웹 애플리케이션 생성하기

5-1) 웹 어플리케이션의 기본 구조



\*톰캣에서 Context는 web application에 해당?

5-2) 웹 애플리케이션 디렉토리(test) 생성 – WEB-INF, META-INF 디렉토리 생성, TEST 폴더내 시작페이지 index.jsp 파일 생성 – web-inf 디렉토리내 classes, lib 디렉토리 생성 – ROOT/ web-inf/web.xml 파일 복사 – test 폴더내 web-inf 디렉토리안에 파일 붙여넣기 – 톰캣 서버 재실행 – localhostL8080/text로 접속하여 새로 만든 웹 애플리케이션 확인 (text web application)

5-3) 톰캣 웹 어플리케이션의 웹 경로 변경하기

Host의 appBase의 하위 디렉토리 확인

Server.xml – 맨 밑 <valve className>태그 밑에

<Context docBase=”test” (웹 어플리케이션 명) path=”webtest” (웹 경로) /> 내용 추가

* Localhost:8080/test/ 에 접속해도, localhost:8080/webtest에 접속해도 같은 애플리케이션 (test web application) 실행됨 (webapps의 하위 디렉토리로 생성된 웹 애플리케이션은 기존 웹 경로와 변경한 웹 경로로 둘 다 접속이 가능)
* 이 방법은 기존 웹 경로, 변경된 웹 경로로 둘 다 접근 가능하므로 권장하는 방법이 아니고 server.xml을 변경해야 하므로 권장하지 않음.

5-4) 톰캣 웹 어플리케이션을 웹 루트 디렉토리로 변경하기

Server.xml – 맨 밑 Host 하위 요소 <valve className>태그 밑에서 path를 기본 웹 경로 (test)에서 웹 루트로 변경 (/) <Context path="/" docBase="test" />

* Webapps의 하위 디렉토리로 생성된 웹 애플리케이션은 웹 루트로 변경해도 기존 웹 경로로도 접근 가능 (권장하지 않음)

5-5) 톰캣 임의의 경로에 웹 애플리케이션 생성하기

<Host> 하위 요소인 <Context>를 추가하여 등록

<Context docBase="C:\JSP\web01" path="web01"/> docBase는 애플리케이션 위치, path는 웹 접근시 경로(주로 웹 어플리케이션명을 부여)

확인: [http://localhost:8080/web01 /](http://localhost:8080/web01%20/) <http://127.0.0.1:8080/web01/>

5-6) 톰캣 임의의 경로에 웹 애플리케이션 생성하기2 [키워드: autoDeploy, Catalina]

Server.xml, <Host name="localhost" autoDeploy="true" unpackWARs="true" appBase="webapps">

autoDeploy 속성 값이 true이면 서버 실행 중에도 변경된 내용이 있으면 자동으로 다시 적재.

* 임의의 경로에 웹 어플리케이션을 생성하고, conf/Catalina/localhost/경로에 (웹 어플리케이션 경로).xml 파일을 작성하여 등록: <Context docBase = “C:\JSP\web02” /> (web02의 경로)
* catalina localhost에 등록된 웹 애플리케이션의 context를 관리. Catalina는 엔진 이름, localhost는 호스트 이름이자 catalina의 defaulthost

<http://localhost:8080/web02> 를 통해 내용 확인

5-7) 톰캣 임의의 경로에 웹 어플리케이션, 웹 경로 변경하기

Catalina – localhost 내에 경로.xml파일에서 파일명을 바꾸고자 하는 경로와 동일하게 바꿔줌.