

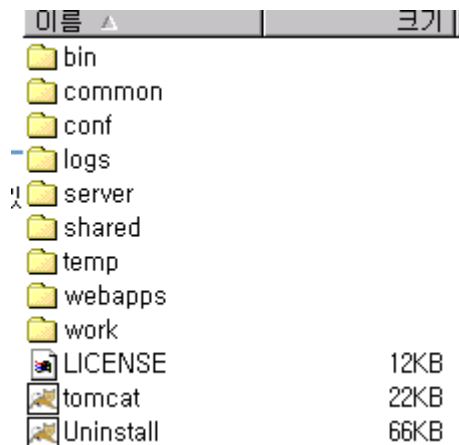
## # Tomcat 디렉토리구조

### 1. CATALINA\_HOME(TOMCAT\_HOME)위치

C:\JAVA\_BIGDATA\program\apache-tomcat-9.0.142.

### TOMCAT\_HOME 디렉토리 구조

(C:\JAVA\_BIGDATA\program\apache-tomcat-9.0.14)



#### - TOMCAT\_HOME /bin

#프로그램의 시작과종료에관련된파일

startup.bat, shutdown.bat

#### - TOMCAT\_HOME /conf

# server.xml

server 의 설정관련파일 (port 설정, host 설정)

ex>port 변경     server.xml LINE:69

-----  
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->

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

connectionTimeout="20000"

redirectPort="8443" />

-----

# web.xml

Web Appliaction에관련된 전역 설정파일

ex> MIME TYPE

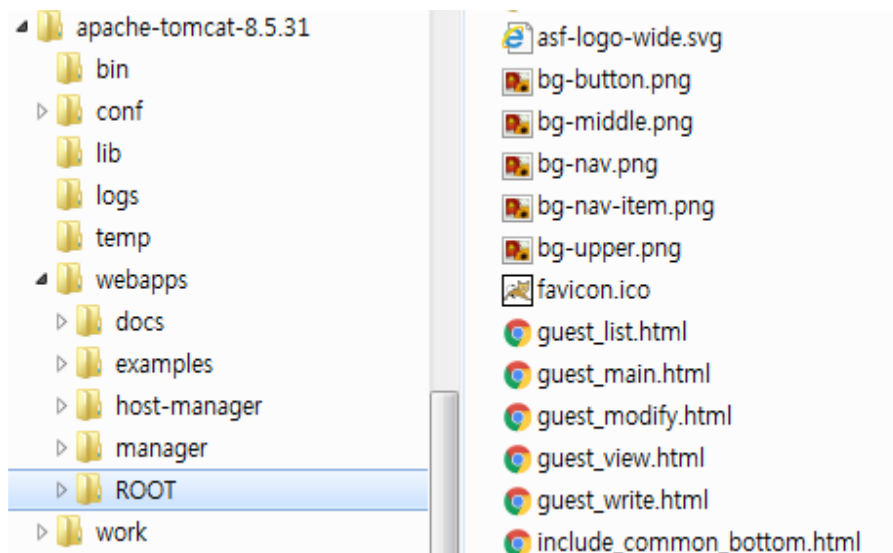
ex> servlet/JSP 실행에 관련된 설정

- TOMCAT\_HOME /webapps

# 웹애플리케이션들이 위치하는 장소

(웹애플리케이션:하나의 사이트,context)

### 3. 웹애플리케이션디렉토리의 구조



-ROOT(웹애플리케이션 이름, context)

\* 클라이언트에 서비스할리소스들이 위치한다.

ex> jsp,gif,html,js,swf .....

- ROOT/WEB-INF

\* web.xml ==> 해당사이트(context) 에만 적용되는 설정

- ROOT/WEB-INF/classes ==> 자바 클래스파일 위치

- ROOT/WEB-INF/lib ==> class 의 묶음 위치(jar..)

#### 4. 웹애플리케이션의 리소스에 접근하는 방법 (URI)

-형식

ex> http://211.254.138.133:80/웹애플리케이션

이름(context이름)/리소스이름

ex> http://211.254.138.133:80/리소스이름

(ROOT 애플리케이션의 경우에는 context 이름인 ROOT 를 생략한다.)

#### 5. 웹애플리케이션의 생성(webapps 안에)

웹애플리케이션이 될려면 디렉토리구조를 만들어줘야한다.

ex> javaspring 이름의 웹애플리케이션의 생성

1. TOMCAT\_HOME /webapps/javaspring directory 생성
2. TOMCAT\_HOME /webapps/javaspring/WEB-INF directory 생성
3. TOMCAT\_HOME /webapps/javaspring/WEB-INF  
안에다가 web.xml file 생성(복사)
4. 요청(http://211.254.138.133/javaspring/xxx.html)

#### 6. 웹애플리케이션의 생성(webapps 밖에)

ex> hmarket 이름의 웹애플리케이션의 생성

- 1 . C:\ JAVA\_BIGDATA \04.WEB\hmarket directory 생성
- 2 . C:\ JAVA\_BIGDATA \04.WEB\hmarket\WEB-INF directory 생성
- 3 . C:\JAVAIOT\04.WEB\hmarket\WEB-INF 안에 web.xml file 생성(복사)
- 4 . TOMCAT\_HOME\conf\ 디렉토리안에 server.xml

server.xml

---

<Host>

.....

<Context

path="/gmarket"

docBase="C:\JAVA\_PYTHON\05.WEB\hmarket"

privileged="true"

reloadable="true"/>

</Host>

---

5. 요청(<http://127.0.0.1/gmarket/index.html>)