


개발환경 02

아파치

Application Developer's Guide

This manual includes contributions from many members of the Tomcat Project developer community. The following authors have provided significant content: The information presented is divided into the following sections: Introduction - Briefly describes the information covered here, with links and references to other

 <https://tomcat.apache.org/tomcat-9.0-doc/appdev/index.html>

Apache Tomcat 9

This is the top-level entry point of the documentation bundle for the Apache Tomcat Servlet/JSP container. Apache Tomcat version 9.0 implements the Servlet 4.0 and JavaServer Pages 2.3 specifications from the Java Community Process, and includes many additional features that make it a useful platform for

 <https://tomcat.apache.org/tomcat-9.0-doc/index.html>

환경설정

export CATALINA_HOME=내 경로/dev/apps/apachetomcat-8.5.56

export PATH = 내 경로:설치디렉토리/bin

1. 톰캣 버전을 선택한다
2. 윈도우 환경 세팅 catalina_home과 path를 잡는다.
3. catalina 환경설정
4. catalina.bat start로 서버 실행
5. catalina.bat stop으로 서버 멈춤
6. webApps에 폴더를 만들고 애플리케이션을 업로드 하면 자동으로 서버에 올라감?(이거는 버전마다 상이함. 9.0.71에서는 가능)

아파치 웹 서버(Apache HTTP Server)의 설정 파일은 "httpd.conf"이다.

SERVER	전체 컨테이너를 대표함
--------	--------------

SERVICE	N개의 CONNECTOR와 엔진을 묶는 SERVER 하위의 컴포넌트
Connector	클라이언트와 통신을 담당하는 여러 커넥터를 사용할 수 있는데 HTTP connector은 http 통신을 담당하고 톰캣 단독 실행시 웹서버의 역할을 한다. Ajp connector은 톰캣과 웹서버를 연결하는 웹서버를 앞에 두고 톰캣을 운영할 경우에 사용된다.
Engine	커넥터로부터 돌아오는 요청을 처리하고 응답을 돌려주는 역할
host	네트워크 이름과 톰캣과의 연결
context	웹 애플리케이션을 의미한다. 여기서 conf/context.xml에 정의되는 것은 웹 애플리케이션 전체에 적용되는 컨텍스트이며 개별 컨텍스트 요소는 명시적으로 정의될 수 있다.

settings.xml의 mirror

```
<mirror>
<id>mirrorId</id>
<mirrorOf>repositoryId</mirrorOf>
<name>Human Readable Name for this Mirror.</name>
<url>http://my.repository.com/repo/path</url>
</mirror>
```

catalina	아파치 톰캣 자체를 의미함. 즉 코어 컴포넌트를 담당한다. 아파치를 가동시키는 것이 카탈리나를 구동시키는 것과 같다는 것을 의미한다.
catalina home	카탈리나 홈은 톰캣 웹 서버의 디렉토리이며 베이스로서 작동한다. 서버를 구동하기 위한 기본 파일들을 애가 다 갖고 있는데 bat파일을 기동했을 때 베이스로서 작동하는 것이다.
catalina base	한 서버에 여러 개의 애플리케이션을 실행하기 위한 디렉토리이다.
type catalina.bat	파일 확인
catalina.bat run	카탈리나(톰캣) 구동
catalina bat start	run과 같음. 창을 끄면 그대로 서버도 내려간다.
mvn clean package -pl 컴파일할 파일 이름 -Dmaven.test.skip=true	특정 프로젝트를 대상으로 실행하겠다.
catalina base	하나의 카탈리나 베이스 서버에 여러개의 웹 애플리케이션을 배치하기 위한 것

** 카탈리나 베이스에 대해서

이클립스에서 서버를 생성하고 프로젝트를 배포해도 아파치 서버 자체에서 또 프로젝트를

배포하면 그 설정이 모두 달라지는 것이다. 이것이 바로 카탈리나 베이스이다 즉 서버를 하나 생성할때마다 servers가 생기는 것처럼 그 세팅이 모두 바뀌어지는 것이다. 톰캣 서버가 실제로 다큐베이스를 통해서 설치되어 있는 카탈리나를 가지고 소스를 어디에 배포하느냐에 따라서 설정이 틀려진다는 것이다. 그러나 같은 프로젝트를 복사해서 배포하면 설정은 같다.

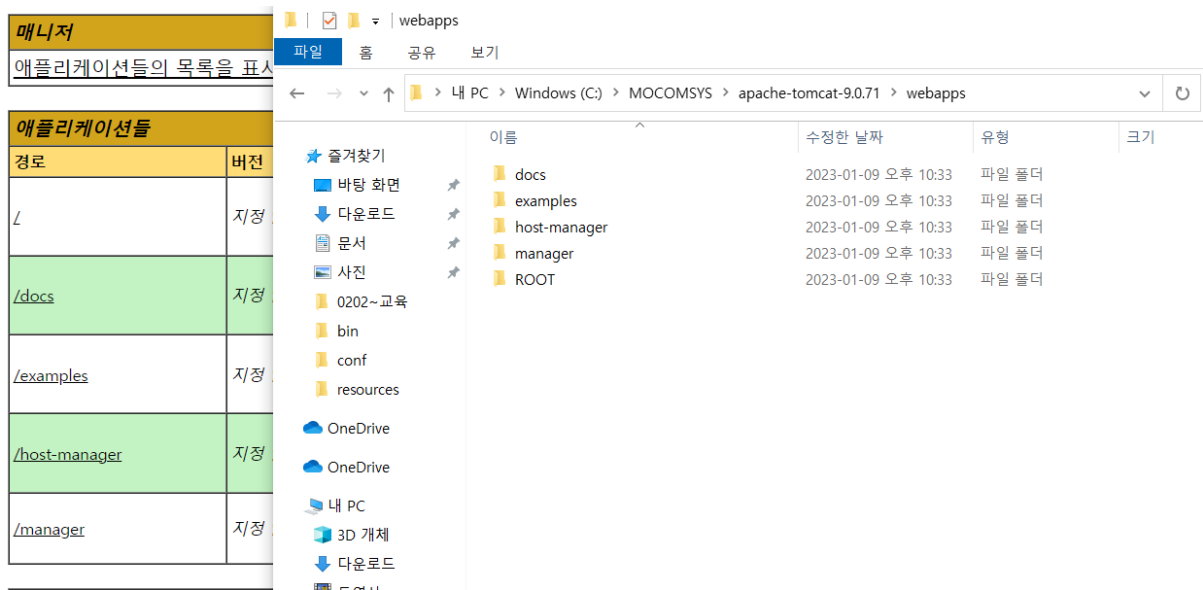
docker run	최초 생성할 때만 이용해야하며 이미 생성된 이름으로 또 생성하면 오류가 난다
docker exec -it maven-example bash	도커를 실행한다
docker run -it --name maven-env -v "pwd":/user/src/mymaven -v "home/.m2":root/.m2 -w /usr/src/mymaven maven:3.3-jdk-8 bash	빌드할 프로젝트 폴더로 이동하고 이동한 현재 디렉토리를 도커볼륨에 매핑하여 메이븐 이미지를 이용하여 컨테이너를 실행하는 명령어
stop run	stop run을 하더라도 이 안에서 만들어진 파일은 다 남아있음 이때 삭제되면 내 로컬에는 남아있지만 컨테이너 자체는 사라지니 컨테이너 안 파일은 삭제된다.
mymvn	이미지 내에 원래부터 존재하는 폴더이며 즉, 컨테이너를 삭제해도 애는 default이다.
docker stop maven-mvn	컨테이너를 멈추고
docker start maven-mvn	컨테이너를 시작한다
docker exec -it maven-mvn bash	컨테이너에 접속하여 mvn clean package로 다시 배포하고 java -cp로 실행을 한다.

apache 폴더 구조

내 PC > Windows (C:) > MOCOMSYS > apache-tomcat-8.5.84			
이름	수정한 날짜	유형	
attachFiles	2023-02-01 오후 1:30	파일 폴더	
bin	2023-01-19 오후 1:11	파일 폴더	
conf	2023-01-19 오후 1:11	파일 폴더	
lib	2023-02-01 오전 11:22	파일 폴더	
logback	2023-02-01 오후 1:41	파일 폴더	
logs	2023-02-01 오후 1:49	파일 폴더	
temp	2023-01-19 오후 1:11	파일 폴더	
webapps	2023-02-22 오후 2:41	파일 폴더	
work	2022-11-16 오후 1:34	파일 폴더	
BUILDING	2023-01-19 오후 1:11	텍스트 문서	

1. bin 폴더: 톰캣 서버의 실행 파일들이 위치한 폴더입니다. startup.bat 또는 startup.sh 파일을 실행하여 톰캣 서버를 시작할 수 있습니다. shutdown.bat 또는 shutdown.sh 파일을 실행하여 톰캣 서버를 중지할 수 있습니다.
2. conf 폴더: 톰캣 서버의 설정 파일들이 위치한 폴더입니다. server.xml 파일은 톰캣 서버의 기본 설정 파일로, 포트번호, SSL 인증서 등을 설정할 수 있습니다. web.xml 파일은 웹 애플리케이션의 설정 파일로, 서블릿 매핑, 보안 설정, 에러 페이지 설정 등을 할 수 있습니다.
3. lib 폴더: 톰캣 서버가 사용하는 라이브러리 파일들이 위치한 폴더입니다. 톰캣 서버 실행 시 필요한 라이브러리들이 lib 폴더에 저장되어 있습니다.
4. logs 폴더: 톰캣 서버의 로그 파일들이 위치한 폴더입니다. 톰캣 서버의 실행 로그, 접근 로그 등이 기록됩니다.
5. temp 폴더: 톰캣 서버가 사용하는 임시 파일들이 위치한 폴더입니다. 예를 들어, JSP 컴파일된 결과 파일 등이 저장됩니다.
6. webapps 폴더: 톰캣 서버에서 실행할 웹 애플리케이션 파일들이 위치한 폴더입니다. 이 폴더에 웹 애플리케이션 파일(war, jar, ear 등)을 배포하면 톰캣 서버에서 해당 웹 애플리케이션을 실행할 수 있습니다. 웹 애플리케이션 파일을 배포하면 webapps 폴더에 해당 웹 애플리케이션의 이름과 동일한 폴더가 생성되고, 그 안에 웹 애플리케이션 파일의 내용이 풀리게 됩니다.
7. work 폴더: 톰캣 서버에서 실행 중인 웹 애플리케이션의 컴파일 결과 파일들이 위치

webapps 애플리케이션 배포하는 폴더

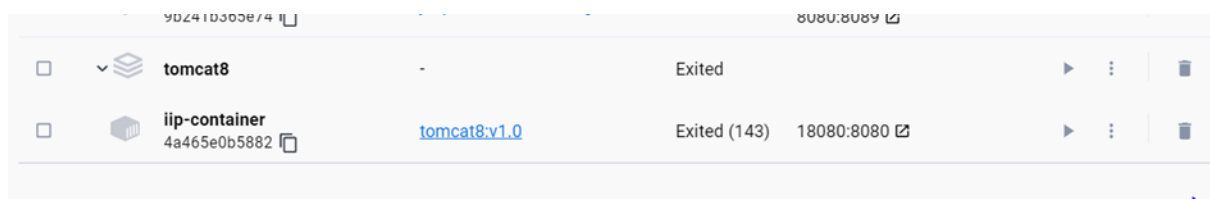


도커와 아파치

Docker-compose

복수 개의 도커 컨테이너를 정의하고 실행하기 위한 도구이다. yaml 파일 형식으로 작성된 설정 파일을 사용하면 여러 개의 컨테이너를 한 번에 실행하고 중지하고 삭제할 수 있는 것이다. 여러 개의 컨테이너를 실행할 때 동일한 네트워크 상에서 동작하도록 설정할 수 있는 것인데 yaml 파일을 형식으로 컨테이너들의 설정을 정의해야 한다. 즉 도커라는 여러 개의 컨테이너를 관리하는 프로세서가 하나 뜨고 복수개의 컨테이너를 관리한다는 것이다.

Docker-compose up과 docker-compose down 명령어를 사용하여 도커를 실행하고 다운시킬 수 있다.



DockerFile

이미지를 빌드하기 위한 설정 파일이다.

FROM tomcat:8.5.46-jdk8-openjdk

톰캣8로 이미지를 생성하겠다.

RUN apt-get update

운영체제에서 사용 가능한 패키지들과 그 버전에 대한 정보를 업데이트 하라는 명령어이다. 그러니까 설치되어 있는 패키지를 최신으로 업데이트 하는 것이 아닌 설치 가능한 리스트를 업데이트 하는 것이라는 것이다.

RUN apt-get install -y tzdata

한국 시간을 설정하기 위해 기본 이미지에 없는 tzdata를 설치해 주는 것이다. (run은 어떤 것을 설치하겠다는 뜻이다)

ENV TZ=Asia/Seoul

업데이트할거 업데이트 하고 인스톨하고 타임존을 설정한다.

CMD ["catalina.sh", "run"]

실제 도커 컨테이너를 실행할 때 카탈리나 런을 실행하겠다.

```
version: '3.0'
services:
  mint:
    container_name: iip-container
    image: tomcat8:v1.0
    environment:
      - TZ=Asia/Seoul
    volumes:
      - C:\mocomsys\docker-home\tomcat8\logs:\usr\local\tomcat\logs
      - C:\mocomsys\docker-home\tomcat8\upload:\usr\local\tomcat\upload
      - C:\mocomsys\docker-home\tomcat8\logback\logback.xml:\usr\local\tomcat\logback\logback.xml
      - C:\mocomsys\docker-home\tomcat8\bin\catalina.sh:\usr\local\tomcat\bin\catalina.sh
      - C:\mocomsys\docker-home\tomcat8\conf\server.xml:\usr\local\tomcat\conf\server.xml
      - C:\mocomsys\docker-home\tomcat8\conf\context.xml:\usr\local\tomcat\conf\context.xml
      - C:\mocomsys\docker-home\tomcat8\webapps\mint-front-product-3.0.0.war:\usr\local\tomcat\webapps\mint.war
      - C:\mocomsys\docker-home\tomcat8\lib\ojdbc6-11.2.0.3.jar:\usr\local\tomcat\lib\ojdbc6-11.2.0.3.jar
    ports:
      - 18080:8080
```

Docker-compose.yaml 파일의 구성이다.

Container-name

컨테이너의 이름을 설정하는 것이다. iip 컨테이너라는 컨테이너를 띄운다는 것.

Image

이미지는 톰캣8 0.1로 띄우겠다는 것이다.

Environment

환경을 설정하는 것인데 타임존은 아시아 서울로 하겠다는 것이다.

Volumes

왼쪽은 내 pc이고 오른쪽은 도커 내부의 경로를 의미한다.

즉, 로컬 환경은 도커에 놔두고 실제 서버는 리눅스에서 돌리는 것을 의미한다. 이미지에서

내려

도커의 원래 환경에 내 pc의 환경을 덮어 씌우겠다는 의미이다. Catalina.sh가 있다면 내 pc에 있는 카탈리나.sh를 서버에 덮어 씌운다는 의미이다.

카탈리나는 수정을 하지 않아도 된다. 리눅스 서버에서 돌아가는 것이기 때문이다.

Ports

왼쪽이 로컬 포트고 오른쪽이 리눅스의 포트다. 리눅스의 포트를 내 포트에 덮어씌운다는 의미이며 접속할 때 포트 번호는 내가 덮어씌운 포트에 접속을 하는 것이다.

도커 내부로 들어가는 명령어(컨테이너 접속)

`docker exec -it`

`beb3d699162cea9d251882c3894eec00d08f4e8ab385b8dab67c8c648a3b057a`

`/bin/bash`

1. **JNDI Resources** - Configuring standard and custom resources in the JNDI naming context that is provided to each web application.
2. **JDBC DataSource** - Configuring a JNDI DataSource with a DB connection pool. Examples for many popular databases.

두개는 더 공부해보기