



Springboot에서 jar에서 war 파일을 만들게 설정 변경하기

OCTOBER 06, 2018

🕒 Reading time ~2 minutes

JAR -> WAR로 배포파일 변경하기

간혹가다 프로젝트를 진행하면서 jar로 배포를 하려다가 war 파일을 톰캣에 올려서 배포하게 변경되는 경우가 생기기도 합니다.

그런 경우 프로젝트가 정말 작고, 시작한 지 얼마 안 된 경우라면 새로 프로젝트를 패키징 설정을 war로 해서 새로 만드는 방법도 있지만... 좋은 방법도 아니고 일정 이상 프로젝트를 진행해서 그러기 쉽지 않은 경우가 발생할 수 있습니다.

그럴 경우 jar 파일로 생성되게 만든 프로젝트를 war 파일을 생성하게 하는 방법에 대해서 알아보시다.

추가로 스프링 부트 2.0.5 버전에서 진행 한 내용이며 실습 코드 자체는 아주 간단하게도 아래와 같이 우리에게 친숙한 hello world를 보여주는 웹 앱입니다.

 localhost:8080/maven2/

hello world!!

일단 jar로 패키징 되게 프로젝트를 만들어 봅니다.

Group:	<input type="text" value="com.example"/>
Artifact:	<input type="text" value="demo"/>
Type:	<input type="text" value="Maven Project (Generate a Maven based project archive)"/>
Language:	<input type="text" value="Java"/>
Packaging:	<input type="text" value="jar"/>
Java Version:	<input type="text" value="8"/>
Version:	<input type="text" value="0.0.1-SNAPSHOT"/>
Name:	<input type="text" value="demo"/>
Description:	<input type="text" value="Demo project for Spring Boot"/>
Package:	<input type="text" value="com.example.demo"/>

일단 maven의 경우 pom.xml의 내용을 보시면 아래와 같이 `packaging` 의 내용에 `jar` 가 들어 있는 걸 확인할 수 있습니다.

```
<groupId>com.example</groupId>
<artifactId>maven</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>
```

gradle의 경우 별다른 내용이 없습니다...

```
buildscript {
    ext {
        springBootVersion = '2.0.5.RELEASE'
    }
    repositories {
        mavenCentral()
    }
    dependencies {
```

```

        classpath("org.springframework.boot:spring-boot-gradle-plugin:${springBootVersion}")
    }

}

apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'org.springframework.boot'
apply plugin: 'io.spring.dependency-management'

```

이 상태에서 프로젝트를 만든 다음 빌드를 해보면 당연히게도 `xxx.jar` 파일이 만들어집니다.

여기서 war 파일을 만드는 것은 아주 간단합니다. 메이븐의 경우

```

<groupId>com.example</groupId>
<artifactId>maven</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>

```

packaging의 내용을 war로 바꿔주기만 하면 됩니다.....

gradle의 경우 build.gradle을 보면

```

apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'org.springframework.boot'
apply plugin: 'io.spring.dependency-management'
apply plugin: 'war'

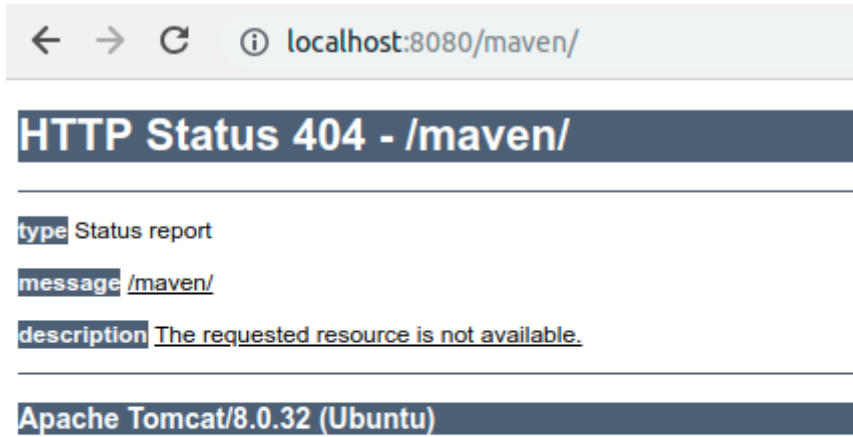
war {
    baseName = 'gradle'
    version = '0.0.1'
}

```

`apply plugin: 'war'` 로 플러그인을 추가하고 `war{}` 에 필요한 내용을 채워 넣으면 됩니다.

그러면 새로 빌드를 하면 `xxx.war` 가 생성이 됩니다. gradle의 경우 위와 같이 설정해주면 `gradle-0.0.1.war` 같이 파일명과 버전링을 해줄 것이라 예상하시겠지만, 그렇지 않으신 분들도 계실 겁니다... (특히 부트 2버전 대가 아니시라면) 그에 대해서는 차후 소개해드리도록 하겠습니다 :)

자 그러면 war 파일도 만들어 봤으니 톱캣을 이용해 배포해 보겠습니다!



?!

여기서 멘붕에 빠지는 경우들이 있으신데... 특히 저 같은 경우 aws에 엘라스틱빈스톡의 톰캣 환경에서 배포를 하기 위해 jar에서 war로 변경했던지라, 스프링 내에 설정을 잘 못 만든 것인지 빈스톡 설정을 잘 못 한 건지 엄청난 멘붕에 빠졌었던 기억이 나는군요.... 결국 당시엔 새로 프로젝트를 만들었습니다..

ServletInitializer

스프링부트를 통해 만든 웹 애플리케이션의 경우 embeded 톰캣을 이용한다면

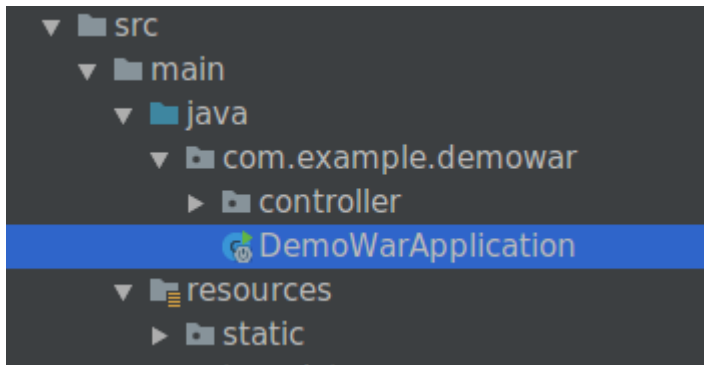
`@SpringBootApplication` 을 구현한 클래스만 있으면 되지만, 따로 톰캣을 통해 배포하는 경우 `SpringBootServletInitializer` 를 상속받은 클래스가 필요합니다.

```
public class ServletInitializer extends SpringBootServletInitializer {

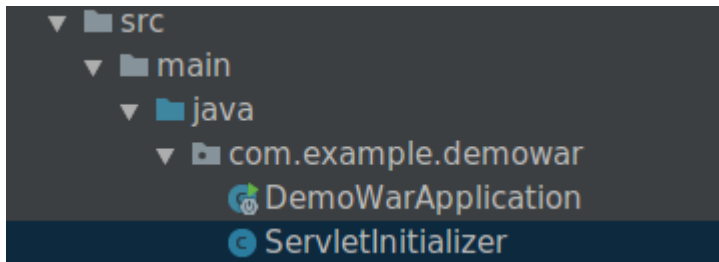
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(DemoWarApplication.class);
    }

}
```

아주 혹시나 알려드리면 `DemoWarApplication.class` 부분은 본인의 `@SpringBootApplication` 구현체를 넣으시면 됩니다. (스프링부트 생성시 존재하는 클래스)

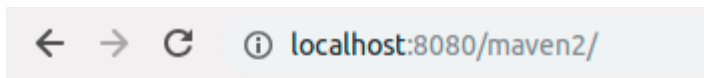


`ServletInitializer` 추가



위와 같이 `ServletInitializer` 를 추가해 주시면 됩니다.(war 패키징 되는 프로젝트로 생성하면 처음 초기화 시 위와 같이 `ServletInitializer`도 같이 생성되는 걸 볼 수 있습니다.)

그리고서 다시 war 파일을 빌드 해주시고 톰캣을 통해 배포하시면



hello world!!

정상적으로 배포되신 것을 확인하실 수 있습니다.

한 줄로 요약하면 maven, gradle에 해당하는 설정 파일에 war 관련 내용을 추가 한 후에, `SpringBootServletInitializer`를 상속 한 `ServletInitializer`를 만들어 주시면 됩니다.

참고 자료 : Spring Boot에서 배포환경 나누기, Spring Boot 웹 애플리케이션을 WAR로 배포할 때 왜 `SpringBootServletInitializer`를 상속해야 하는걸까?

(`SpringBootServletInitializer`에 대해 보다 자세히 알고 싶으시다면 두 번째 참고 자료를 추천드립니다.)

SPRING WAR



LIKE



TWEET



+1

댓글 0건

hue9010

로그인 ▾

추천

Tweet

공유

인기순 ▾

토론 시작

다음으로 로그인

또는 디스커스에 가입하세요.

이름

1등으로 댓글 달기

HUE9010의 다른 댓글.

171219-TIL – 폭군길냥의 블로그

댓글 2건 • 2년 전

 아바타 이학선 — 플러그인 있을거라고
타 생각을 안해봤었는데!!!
덕분에 해결 했습니다. 밥 드시**2017년을 회고하며 – 폭군길냥의 블로그**

댓글 7건 • 2년 전

 아바타 이학선 — 이동진씨가 누군가 고
타 민 했는데....
반갑습니다. 동진씨ㅋㅋㅋㅋ**180123-TIL – 폭군길냥의 블로그**

댓글 한 건 • 2년 전

아바타 will Park — 뽕 ~.~

171130-TIL – 폭군길냥의 블로그

댓글 한 건 • 2년 전

 아바타 정휘준 — 음, ForwardController
타 클래스가 구조적으로 한 개의

Read More

사용한 명령어들의 기록을 보는 history

history 명령어에 대해서 알아보자 [Continue reading](#)

autowired로 의존성을 주입하는 세 가지 방법

Published on January 27, 2019

netstat, lsof를 통해 특정 포트(port)를 사용 중인 프로세스의 pid 찾기

Published on January 20, 2019

© 2019 hue9010. Powered by Jekyll using the HPSTR Theme.