

Gradle: Jar

jar 안에는 META-INF/MANIFEST.MF 파일에 실행할 main() 메서드의 클래스를 지정

META-INF/MANIFEST.MF

Manifest-Version: 1.0

Main-Class: hello.embed.EmbedTomcatSpringMain

Jar 파일 문제

- Jar파일은 Jar파일을 포함 할 수 없다.
- 따라서, Jar파일 안에 라이브러리가 없으므로 Jar파일을 실행할 수 없다.

해결 방법

- Class file은 포함할 수 있다.
- 따라서, jar를 다 풀어서 Class file 형태로 가지고 있자! → Fat Jar

Gradle: Fat Jar or Uber Jar

필요한 라이브러리들을 모두 Class로 풀어서 가지고 있게 된다.

Fat Jar 파일 문제

- Jar안에 모든 파일을 풀어놓았다.
- 따라서, 파일명 중복이 발생하는 경우, 충돌이 발생하게 된다.

문제 해결

- 이러한 문제를 해결하기 위해 실행가능한 Jar를 지원한다.

Gradle: Executable Jar

META-INF/MANIFEST.MF

```
Manifest-Version: 1.0
Main-Class: org.springframework.boot.loader.JarLauncher
Start-Class: hello.boot.BootApplication
Spring-Boot-Version: 3.0.2
Spring-Boot-Classes: BOOT-INF/classes/
Spring-Boot-Lib: BOOT-INF/lib/
Spring-Boot-Classpath-Index: BOOT-INF/classpath.idx
Spring-Boot-Layers-Index: BOOT-INF/layers.idx
Build-Jdk-Spec: 17
```

먼저 실행하기 위한 Jar를 인식한 후,
우리가 정의한 main()을 실행시킨다.

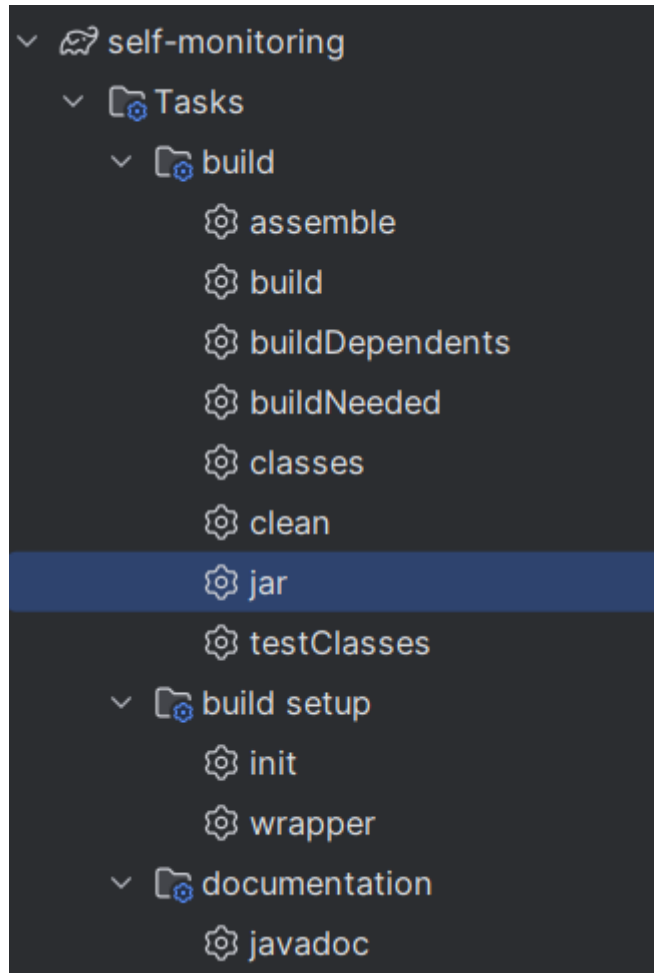
실행 과정 정리

- 1. `java -jar xxx.jar`
- 2. `MANIFEST.MF` 인식
- 3. `JarLauncher.main()` 실행
 - `BOOT-INF/classes/` 인식
 - `BOOT-INF/lib/` 인식
- 4. `BootApplication.main()` 실행

참고

실행 가능 Jar가 아니라, IDE에서 직접 실행할 때는 `BootApplication.main()` 을 바로 실행한다.
IDE가 필요한 라이브러리를 모두 인식할 수 있게 도와주기 때문에 `JarLauncher` 가 필요하지 않다.

Gradle: Executable Jar



Plugin 추가

```
plugins { this: PluginDependenciesSpecScope
  java
  id("org.springframework.boot") version "3.2.1"
```

