

목차

Override logging in WildFly/JBoss	3
방법1: add-logging-api-dependencies 변경	3
방법2: jboss-deployment-structure.xml을 사용	4
참고사항	4
관련링크	5

Override logging in WildFly/JBoss

logback을 사용하여 애플리케이션에서 로깅을 구성 할 수 있습니다. logback을 사용하여 서버에 대한 로깅을 구성 할 수 없습니다.

1. 구성에서 logback을 사용하려면 add-logging-api-dependencies를 false로 변경하거나
2. 하위 시스템을 제외하는 jboss-deployment-structure.xml을 만들어야합니다.
3. 또한 배포에 logback 및 slf4j를 포함시켜야합니다.

방법1: add-logging-api-dependencies 변경

add-logging-api-dependencies를 변경하는 첫 번째 옵션은 모든 배포에 대한 전역 설정입니다. follow CLI 명령은 값을 변경합니다.

```
/subsystem=logging:write-attribute(name=add-logging-api-dependencies,value=false)
```

위 CLI명령어는 standalone.xml 내의 아래부분에 추가됩니다.

```
<subsystem xmlns="urn:jboss:domain:logging:6.0">
  <add-logging-api-dependencies value="false"/>
  <!-- <=>
이부분에 추가됨 -->
  <console-handler name="CONSOLE">
    <level name="INFO"/>
    <formatter>
      <named-formatter name="COLOR-PATTERN"/>
    </formatter>
  </console-handler>
  <periodic-rotating-file-handler name="FILE" autoflush="true">
    <formatter>
      <named-formatter name="PATTERN"/>
    </formatter>
    <file relative-to="jboss.server.log.dir" path="server.log"/>
    <suffix value=".yyyy-MM-dd"/>
    <append value="true"/>
  </periodic-rotating-file-handler>
  <logger category="com.arjuna">
    <level name="WARN"/>
  </logger>
  <logger category="io.jaegertracing.Configuration">
    <level name="WARN"/>
  </logger>
  <logger category="org.jboss.as.config">
    <level name="DEBUG"/>
  </logger>
</subsystem>
```

```

    <logger category="sun.rmi">
      <level name="WARN"/>
    </logger>
    <root-logger>
      <level name="INFO"/>
      <handlers>
        <handler name="CONSOLE"/>
        <handler name="FILE"/>
      </handlers>
    </root-logger>
    <formatter name="PATTERN">
      <pattern-formatter pattern="%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p
[%c] (%t) %s%n"/>
    </formatter>
    <formatter name="COLOR-PATTERN">
      <pattern-formatter pattern="%K{level}%d{HH:mm:ss,SSS} %-5p
[%c] (%t) %s%n"/>
    </formatter>
  </subsystem>

```

이 옵션은 배포에 implicit 로깅 종속성을 추가하지 않습니다.

방법2: jboss-deployment-structure.xml을 사용

jboss-deployment-structure.xml을 사용하는 두 번째 옵션은 배포에 대한 로깅 하위 시스템 만 비활성화합니다. 다음은 예제 파일입니다.

이 파일의 위치는 /META-INF/jboss-deployment-structure.xml에 두어야 적용됩니다.

```

<jboss-deployment-structure>
  <deployment>
    <!-- exclude-subsystem prevents a subsystems deployment unit processors
running on a deployment -->
    <!-- which gives basically the same effect as removing the subsystem,
but it only affects single deployment -->
    <exclude-subsystems>
      <subsystem name="logging" />
    </exclude-subsystems>
  </deployment>
</jboss-deployment-structure>

```

참고사항

- Wildfly / JBoss Application 관련 설정파일 위치는 아래와 같습니다.

```

TEST.ear
|--/META-INF/application.xml
|--/META-INF/jboss-deployment-structure.xml

```

```
| --/TEST.war ==> check this archive file exists
```

- Spring-Boot 사용환경에서는 logback.xml 설정파일의 파일명을 logback-spring.xml 로 하여야 합니다.

관련링크

- <https://stackoverflow.com/questions/21846329/override-logging-in-wildfly>
- <https://access.redhat.com/solutions/4566431>
- <https://coding-start.tistory.com/186>

From:
<https://atl.kr/dokuwiki/> - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/override_logging_in_wildfly_jboss

Last update: **2019/11/27 12:35**

