

Spring vs Springboot



1주차 과제(Spring 환경 설정)와 3주차 과제(Springboot 세팅)을 바탕으로, Spring과 Springboot의 환경 설정 차이를 정리한 내용입니다.

1. Springboot는 의존성 & 버전 관리가 자동화되어 있다.

- Spring은 pom.xml에서 jdbc, JUnit등의 의존성(dependency)을 추가할 때, Spring에서는 각 라이브러리 버전을 상호 호환이 되도록 개발자가 직접 설정해야 했다.

```
<!-- Spring JDBC -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>4.3.9.RELEASE</version>
</dependency>
```

- Springboot는 pom.xml의 <parent>태그에 spring-boot-starter-parent가 dependency의 외부 라이브러리들의 버전과 상호 의존성을 관리한다. 따라서 개발자가 직접 version을 명시하지 않아도, spring-boot-starter-parent에서 선언된 버전이 자동 설정된다.
(→ *spring-boot-starter-parent가 가지고 있지 않은 라이브러리는 버전을 명시해줘야 하며, 포함된 라이브러리에 다른 version을 오버라이딩하는 것도 가능하다.*)

```
<!-- spring-boot-starter-parent 설정 -->
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.2.2.RELEASE</version>
  <relativePath/>
</parent>

<!-- Springboot JDBC -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
```

- 참고: <https://dailyheumsi.tistory.com/166>

2. Springboot는 내장 Tomcat이 있다.

- Spring설정시에는 Tomcat9을 다운받아서 이클립스에서 Tomcat의 경로를 직접 설정하여 서버를 세팅해야 했다.
- Springboot에는 Tomcat 서버가 내장되어 있기 때문에, Web starter 의존성 추가만으로 WAS를 사용할 수 있다.
(→ 내장 Tomcat이 있기 때문에, @ SpringBootApplication 어노테이션이 선언되어 있는 클래스의 main() 메소드를 실행하기만 해도 서버를 구동시킬 수 있다.)

```
<!-- pom.xml -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

3. Springboot 설정은 application.properties 파일에서 설정한다.

- Spring에서는 web.xml, root-context.xml, servlet-context.xml과 같이 각 설정 파일마다 필요한 설정을 해야 했다.
 - web.xml: 인코딩 방식, 서블릿 매핑 등 설정을 위한 설정파일
 - root-context.xml: Service, DB, Repository(DAO)등 비즈니스 로직과 관련된 설정
 - servlet-context.xml: Controller, view resolver 등 요청과 관련된 객체를 정의
 - 참고: <https://hamdi.tistory.com/15>
- Springboot에서는 contextPath, view, DB 등 각종 설정을 application.properties 한 곳에서 설정한다.

```
/* application.properties 예시 */

server.port = 8031
server.servlet.contextPath=/

spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

devtools.livereload.enabled=true

spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
spring.datasource.url=jdbc:mariadb://127.0.0.1:3306/statistic
```

```
spring.datasource.username=root  
spring.datasource.password=123456
```