

49강. Build.gradle 이해하기

build.gradle

빌드 스크립트라고도 불리며,



gradle을 이용해 프로젝트를 빌드하고
의존성을 관리하기 위해 작성되었다.

build.gradle



groovy 언어를 사용해 작성되었고,

kotlin이라는 언어를 사용할 수도 있다.



이제 build.gradle를 하나씩 살펴보자!

plugins 블록

```
plugins {  
    id 'org.springframework.boot' version '2.7.6'  
    id 'io.spring.dependency-management' version '1.0.12.RELEASE'  
    id 'java'  
}
```

플러그인을 추가할 수 있다!

plugins 블록

```
plugins {  
    id 'org.springframework.boot' version '2.7.6'  
    id 'io.spring.dependency-management' version '1.0.12.RELEASE'  
    id 'java'  
}
```

이름 : org.springframework.boot
버전 : 2.7.6

org.springframework.boot 플러그인 역할

- 1) 스프링을 빌드했을 때 실행가능한 jar 파일이 나오게 도와주고
- 2) 스프링 애플리케이션을 실행할 수 있게 도와주고
- 3) 또다른 플러그인들이 잘 적용될 수 있게 해준다.

plugins 블록

```
plugins {  
    id 'org.springframework.boot' version '2.7.6'  
    id 'io.spring.dependency-management' version '1.0.12.RELEASE'  
    id 'java'  
}
```

외부 라이브러리, 프레임워크의 버전관리에 도움을 주고
서로 얹혀 있는 의존성을 처리하는데 도와준다.

plugins 블록

```
plugins {  
    id 'org.springframework.boot' version '2.7.6'  
    id 'io.spring.dependency-management' version '1.0.12.RELEASE'  
    id 'java'  
}
```

Java 프로젝트를 개발하는데 필요한 기능들을 추가해주고,
다른 JVM 언어 Gradle 플러그인을 사용할 수 있는 기반을 마련한다.

다음으로 눈에 들어오는 부분은..!

프로젝트와 관련된 설정

```
group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11'
```

프로젝트의 그룹,
빌드 결과물에 프로젝트 그룹에 대한 정보가 들어 있다.

프로젝트와 관련된 설정

```
group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11'
```

프로젝트의 버전

프로젝트와 관련된 설정

```
group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11'
```



library-app-0.0.1-SNAPSHOT.jar

프로젝트의 버전

프로젝트와 관련된 설정

```
group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11'
```

프로젝트가 사용하고 있는 JDK 버전

repositories 블록

```
repositories {  
    mavenCentral()  
}
```

외부 라이브러리/프레임워크를 가져오는 장소 설정

repositories 블록

```
repositories {  
    mavenCentral()  
}
```

mavenCentral() : maven 중앙 저장소

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

우리가 사용하는 라이브러리/프레임워크를 표시하는 곳

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

org.springframework.boot가 만든
spring-boot-starter-data-jpa를 가져와라!

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

implementation : 해당 의존성을 항상 사용한다.
runtimeOnly : 코드를 실행할 때에만 의존성을 사용한다.

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

testImplementation :
테스트 코드를 컴파일 하거나 실행시킬 때 항상 사용한다

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

Dependency Configuration

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

spring-boot-starter-data-jpa

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

spring-boot-**starter**-data-jpa

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

JPA를 스프링과 함께 사용하기 위해 필요한 모든 것!

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

스프링으로 web 관련 개발을 하기 위해 필요한 모든 것!

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

h2와 mysql을 사용하기 위해 필요한 의존성

dependencies 블록

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    runtimeOnly 'com.h2database:h2'  
    runtimeOnly 'mysql:mysql-connector-java'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

스프링을 사용해 테스트를 작성하기 위해 필요한 라이브러리,
프레임워크가 모여 있는 의존성!

테스트를 수행할 때 Junit5를 사용하겠다!

```
tasks.named('test') {  
    useJUnitPlatform()  
}
```

50강. Spring과 Spring Boot

Spring과 Spring Boot의 간단 역사

JavaEE
디자인 책 발간

2002년
10월



Java EE(Enterprise Edition)으로 서버 개발을 하던 시대...
추운 겨울이었다...!

Spring과 Spring Boot의 간단 역사

JavaEE
디자인 책 발간

2002년
10월



그러다 Rod Johnson 이라는 분이 한 책을 저술하는데...

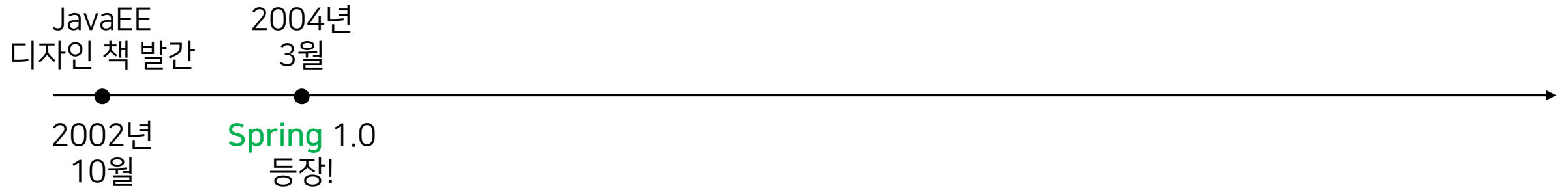
Spring과 Spring Boot의 간단 역사

JavaEE
디자인 책 발간

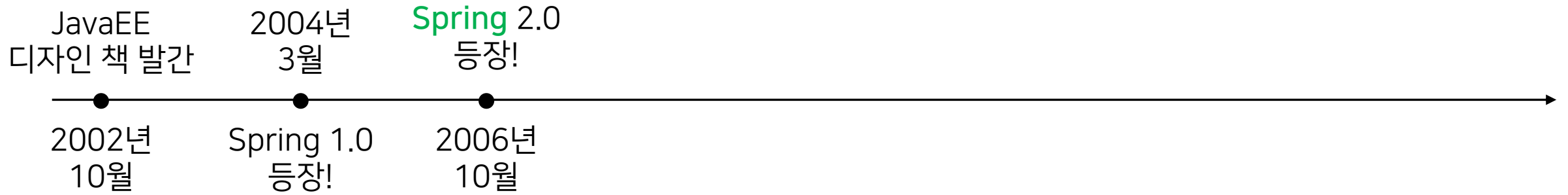
2002년
10월

프로그래밍 업계를 뒤집어 놓았다...!

Spring과 Spring Boot의 간단 역사

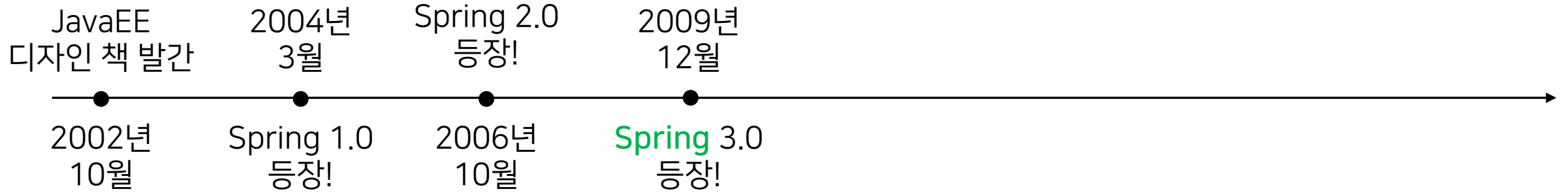


Spring과 Spring Boot의 간단 역사



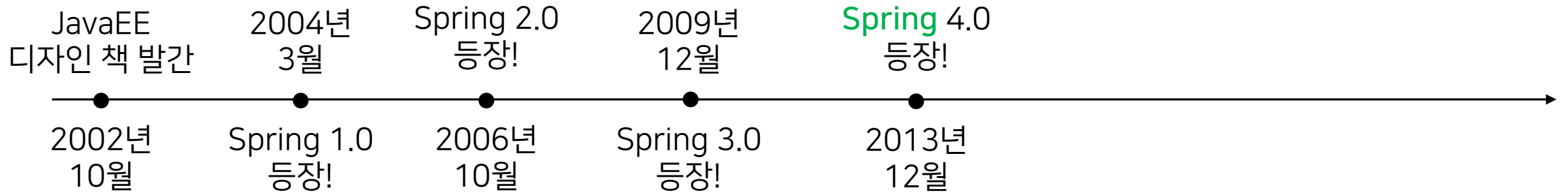
XML 설정이 간단해지고, Java5 지원이 이루어졌다.

Spring과 Spring Boot의 간단 역사



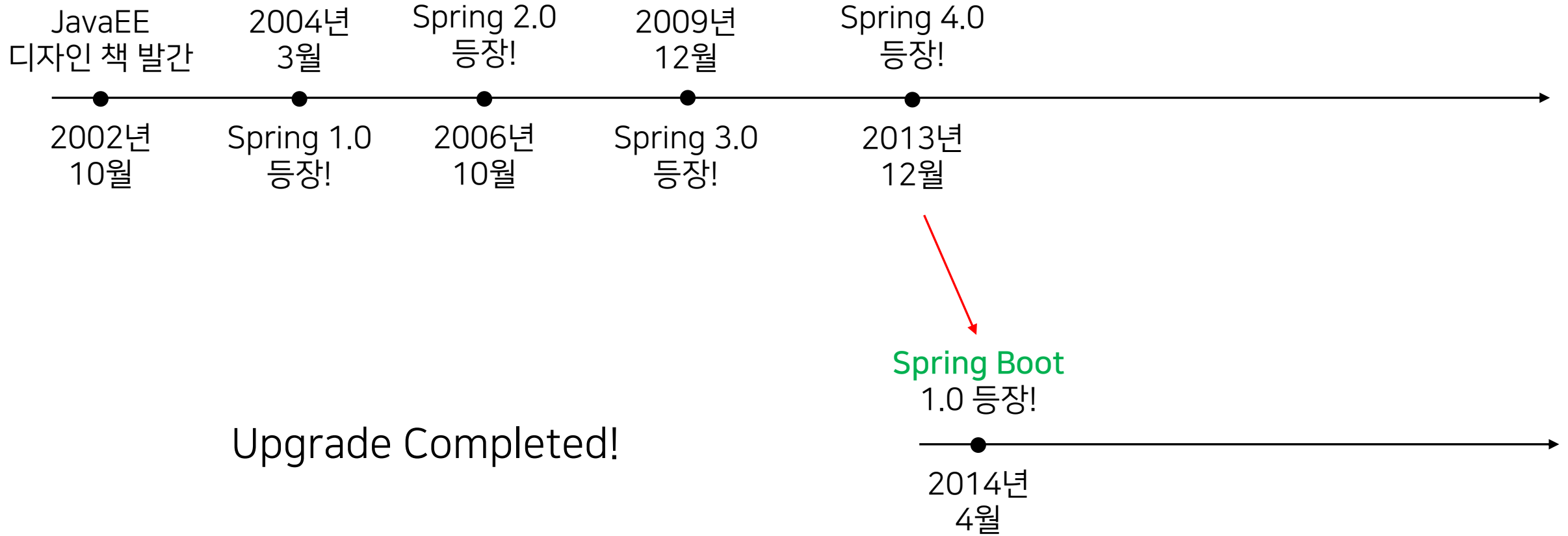
Java 기반 설정 (Annotation) 확대, Java6 지원

Spring과 Spring Boot의 간단 역사

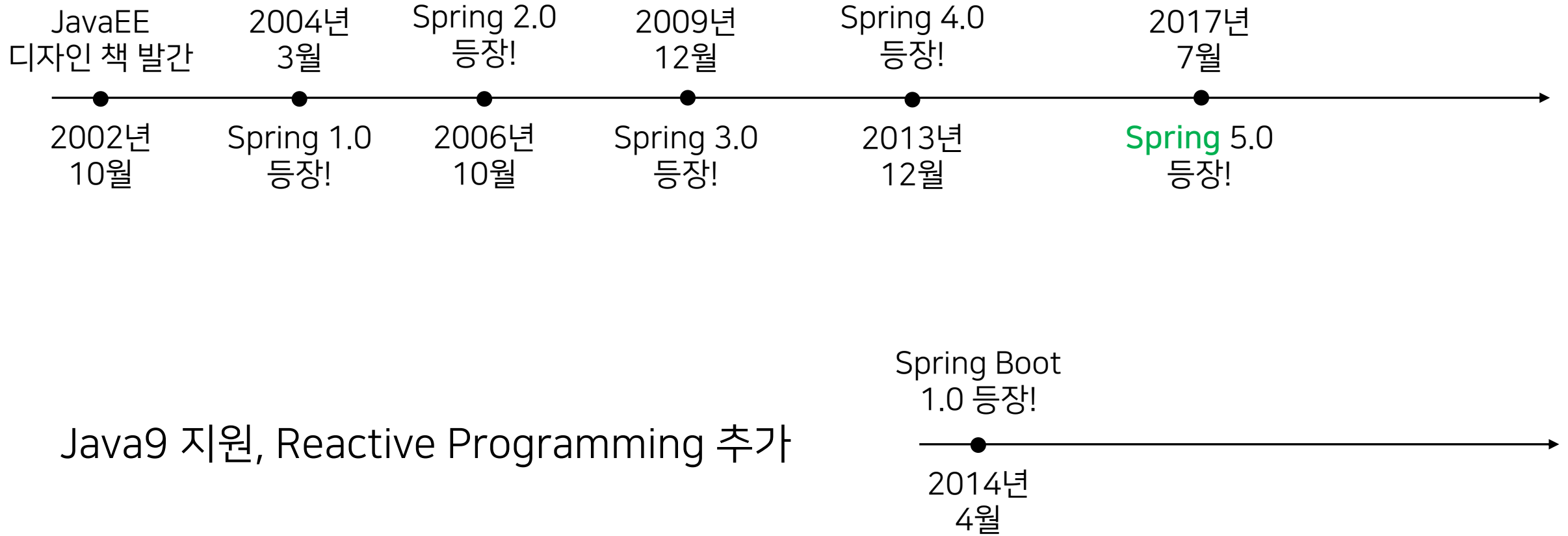


Java8지원, Bean 정의를 위해 groovy DSL 활용, 웹소켓 지원

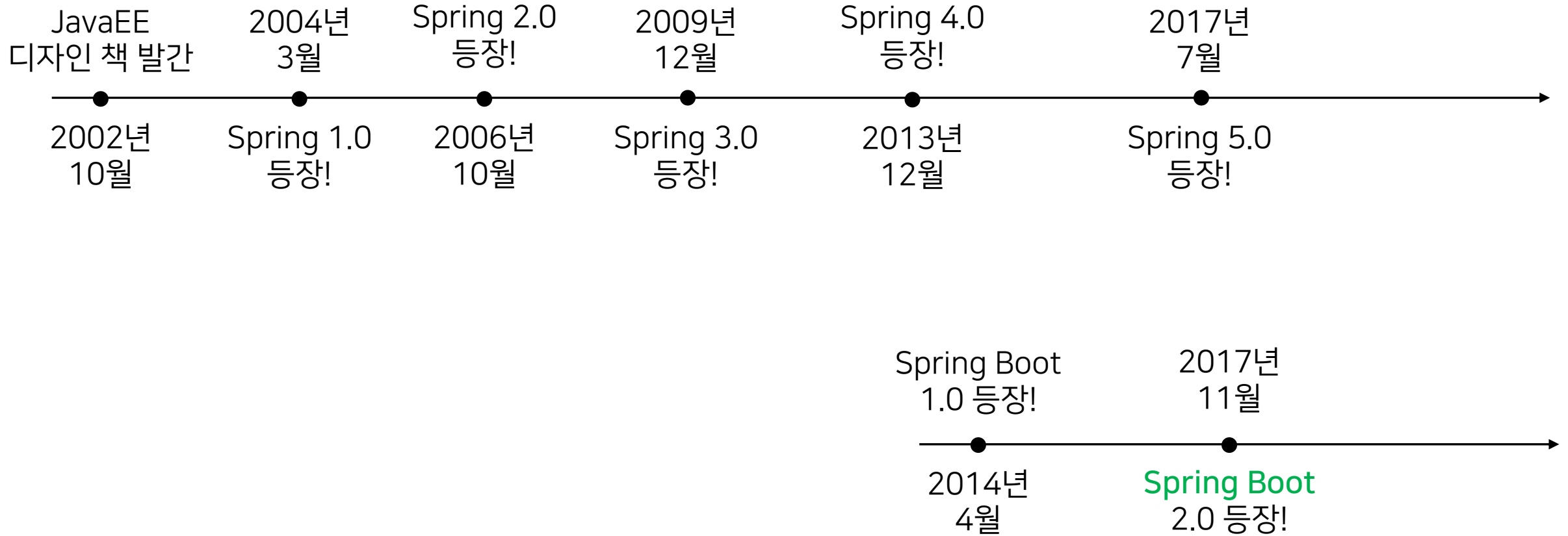
Spring과 Spring Boot의 간단 역사



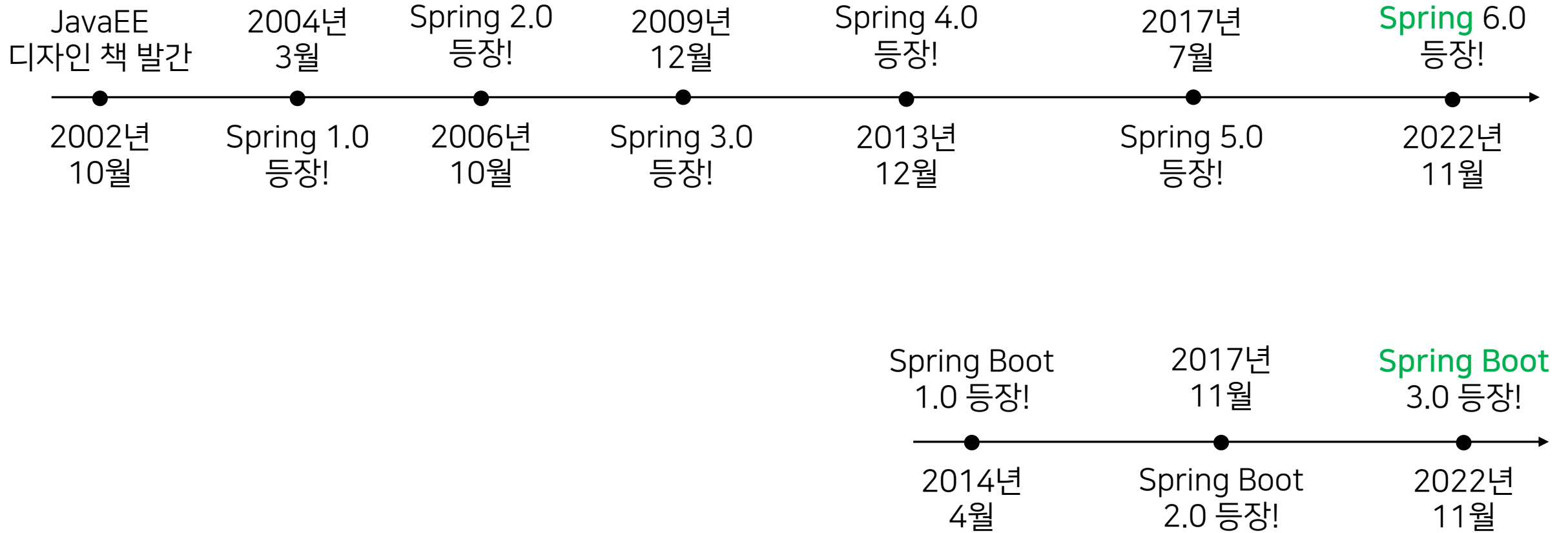
Spring과 Spring Boot의 간단 역사



Spring과 Spring Boot의 간단 역사



Spring과 Spring Boot의 간단 역사



2017년 11월에 등장한 Spring Boot 2.0.x

한글 마이그레이션 자료는?!

2017년 11월에 등장한 Spring Boot 2.0.x

Google

spring boot 2 마이그레이션

전체 동영상 이미지 뉴스 지도 더보기

검색결과 약 5,230,000개 (0.45초)

<https://yonguri.tistory.com> > ...

SpringBoot 1.x -> 2.x 마이그레이션 절차 - 대디장의 기억저장소

2020. 11. 11. — 1.x → 2.x로의 메이저 업데이트이기 때문에 단순히 스프링부트 버전만 올려서 적용하면, 수많은 컴파일エラー와 런타임시의 각종 트러블을 직면하게 됩니다 ...

스프링 소셜에 대한 Auto Configuration 제외: 의... 엘라스틱서치 라이브러리 버전변경: 5.4
JPA 2.0 에서 id Auto_increment 관련 수정: 이슈 - ... 액추에이터 API 변경: 자체 매트릭 API → Micro...

<https://github.com> > spring-projects > spring-boot > wiki

Spring Boot 2.0 Migration Guide - GitHub

2021. 12. 9. — The move to Spring Boot 2 will upgrade a number of dependencies and might require work on your end. You can review dependency management for ...

<https://velog.io> > Spring-SpringBoot-마이그레이션-2탄

Spring → SpringBoot 마이그레이션 2탄 - velog

2022. 2. 21. — 삼질기. 모든 코드는 깃허브에 있다. 일단 저번에 마이그레이션 1탄을 했었는데 의존성이 겹쳐서 다시 리팩토링 하려고 보니까 에러가 많았다.

<https://shanepark.tistory.com> > ...

Spring Boot 1.5 -> 2.5 마이그레이션 회고

2022. 3. 15. — 스프링 부트2 버전으로 업그레이드. dependency 들에 대한 검토를 마쳤다면 2.0.x 중 최신 버전으로 업그레이드 합니다. 2.0.0.

Intro · Before You Start · 스프링 부트2 버전으로 업그레이드... · 프로젝트 적용중 문제 해결

<https://thinkground.studio> > spring-boot-2-migration-스...

Spring Boot 2 Migration (스프링부트 마이그레이션) 검토 내용

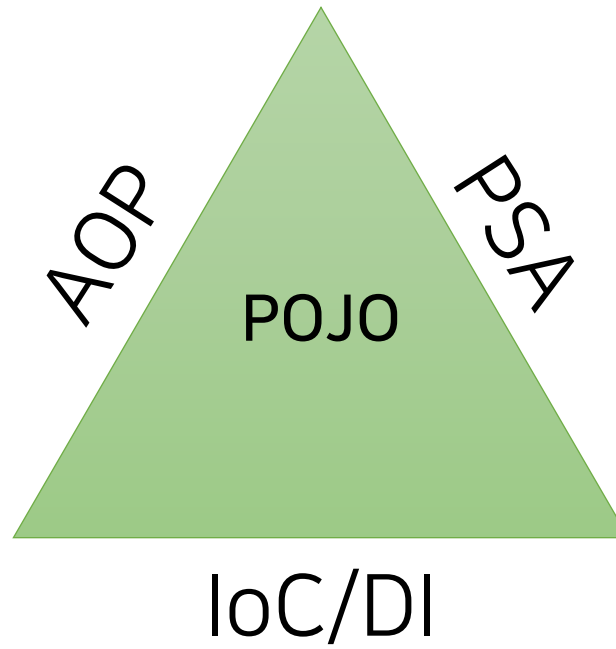
2020. 12. 6. — 최근 스프링 부트를 시작하신 분들은 관련이 없는 내용이지만, 이전에 개발된 솔루션은 아직 Spring Boot의 Major 버전이 1 인 경우가 많습니다. 운영 ...

2020~2021년 자료가 많다!

바로바로 최신화 하는 것이 원래 어렵습니다...!

그렇다면 Spring Boot가 나오며 무엇이 달라졌을까?!

[1] 간편한 설정



스프링은 강력한 기능을 제공한다!

[1] 간편한 설정

하지만, 강력한 기능 사용을 위해 xml 설정을 많이 해야 했다!
많아도... 너무 많다!!!

[1] 간편한 설정



어노테이션 기반의 설정 적극 사용!

기본적으로 필요한 것 모두 자동 설정!

[2] 간단한 의존성 관리

Spring을 사용할 때에는 개발에 필요한 라이브러리/프레임워크를 모두 적어야 했다!

[2] 간단한 의존성 관리

spring-boot-starter-data-jpa

```

└─ org.springframework.boot:spring-boot-starter-data-jpa:2.7.2
   └─ jakarta.persistence:jakarta.persistence-api:2.2.3
      └─ jakarta.transaction:jakarta.transaction-api:1.3.3
         └─ org.hibernate:hibernate-core:5.6.10.Final
            └─ org.springframework.boot:spring-boot-starter-aop:2.7.2
               └─ org.aspectj:aspectjweaver:1.9.7
                  └─ org.springframework.boot:spring-boot-starter:2.7.2
                     └─ jakarta.annotation:jakarta.annotation-api:1.3.5
                        └─ org.springframework.boot:spring-boot-autoconfigure:2.7.2
                           └─ org.springframework.boot:spring-boot:2.7.2 (*)
                              └─ org.springframework.boot:spring-boot-starter-logging:2.7.2
                                 └─ ch.qos.logback:logback-classic:1.2.11
                                    └─ ch.qos.logback:logback-core:1.2.11
                                       └─ org.slf4j:slf4j-api:1.7.36
                                          └─ org.apache.logging.log4j:log4j-to-slf4j:2.17.2
                                             └─ org.apache.logging.log4j:log4j-api:2.17.2
                                                └─ org.slf4j:slf4j-api:1.7.36
                                                   └─ org.slf4j:jul-to-slf4j:1.7.36
                                                      └─ org.slf4j:slf4j-api:1.7.36
                                                         └─ org.springframework.boot:spring-boot:2.7.2
                                                            └─ org.springframework:spring-context:5.3.22
                                                               └─ org.springframework:spring-aop:5.3.22
                                                                  └─ org.springframework:spring-beans:5.3.22
                                                                     └─ org.springframework:spring-core:5.3.22 (*)
                                                                        └─ org.springframework:spring-core:5.3.22 (*)
                                                                           └─ org.springframework:spring-beans:5.3.22 (*)
                                                                              └─ org.springframework:spring-core:5.3.22 (*)
                                                                                 └─ org.springframework:spring-expression:5.3.22
                                                                                    └─ org.springframework:spring-core:5.3.22 (*)
                                                                                       └─ org.springframework:spring-core:5.3.22
                                                                                          └─ org.springframework:spring-jcl:5.3.22
                                                                                             └─ org.springframework:spring-core:5.3.22 (*)
                                                                                                └─ org.yaml:snakeyaml:1.30
                                                                                                   └─ org.springframework:spring-aop:5.3.22 (*)
                                                                                                      └─ org.springframework.boot:spring-boot-starter-jdbc:2.7.2
                                                                                                         └─ com.zaxxer:HikariCP:4.0.3
                                                                                                            └─ org.slf4j:slf4j-api:1.7.36
                                                                                                               └─ org.springframework.boot:spring-boot-starter:2.7.2 (*)
                                                                                                                  └─ org.springframework:spring-jdbc:5.3.22
                                                                                                                     └─ org.springframework:spring-beans:5.3.22 (*)
                                                                                                                        └─ org.springframework:spring-core:5.3.22 (*)
                                                                                                                           └─ org.springframework:spring-tx:5.3.22

```

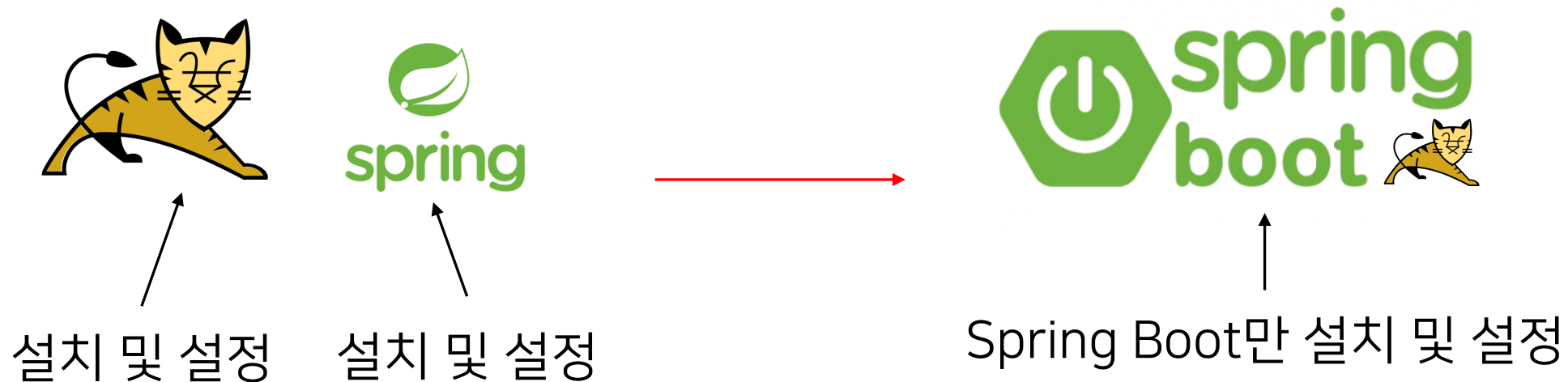
[2] 간단한 의존성 관리



뭐? 의존성 관리가 어려워?!

Starter로 다 묶어줄게!!!

[3] 강력한 확장성



Spring과 Spring Boot의 차이점

1. 간편한 설정
2. 간단한 의존성 관리
3. 강력한 확장성
4. MSA에 적합한 모니터링

**51강. application.yml과 application.properties,
그리고 lombok**

application.yml 파일

YAML 문법

Yet Another Markup Language

또다른 마크업 언어

YAML Ain't Markup Language

YAML은 마크업 언어가 아니다!

```
spring:
  config:
    activate:
      on-profile: local
  datasource:
    url: "jdbc:h2:mem:library;MODE=MYSQL;NON_KEYWORDS=USER"
    username: "sa"
    password: ""
    driver-class-name: org.h2.Driver
  jpa:
    hibernate:
      ddl-auto: create
    properties:
      hibernate:
        show_sql: true
        format_sql: true
        dialect: org.hibernate.dialect.H2Dialect
  h2:
    console:
      enabled: true
      path: /h2-console
```

YAML

YAML 파일의 확장자 : .yaml 또는 .yml

YAML의 간단 문법

```
spring:  
  config:  
    activate:  
      on-profile: dev
```

key: value 형식으로 데이터를 정의한다.

YAML의 간단 문법

```
spring:  
  config:  
    activate:  
      on-profile: dev
```

key : spring.config.active.on-profile / **value** : dev

YAML의 간단 문법

```
spring:  
  datasource:  
    url: "jdbc:mysql://localhost/library"  
    username: "root"  
    password: "Abcd1234!"  
    driver-class-name: com.mysql.cj.jdbc.Driver
```

각 계층은 들여쓰기를 통해 구분하게 되며,
이를 이용해 중복을 제거할 수 있다!

YAML의 간단 문법

```
spring:  
  datasource:  
    url: "jdbc:mysql://localhost/library"  
    username: "root"  
    password: "Abcd1234!"  
    driver-class-name: com.mysql.cj.jdbc.Driver
```

spring.datasource.url
spring.datasource.username

YAML의 간단 문법

value에는 1) 참 / 거짓 2) 숫자 3) 문자열이 들어갈 수 있다!

YAML의 간단 문법

value에는 1) 참 / 거짓 2) 숫자 3) 문자열이 들어갈 수 있다!



true / false

YAML의 간단 문법


value에는 1) 참 / 거짓 2) 숫자 3) 문자열이 들어갈 수 있다!



그냥 숫자

YAML의 간단 문법

value에는 1) 참 / 거짓 2) 숫자 3) 문자열이 들어갈 수 있다!



큰 따옴표가 있어도 되고,
없어도 된다.

YAML의 간단 문법

```
person:  
  name: lannstark  
  age: 99  
  dogs:  
    - 초코  
    - 망고
```

value에는 배열도 들어갈 수 있다! - 를 활용한다.

YAML의 간단 문법

```
# 주석  
spring:  
  config:  
    activate:  
      on-profile: dev
```

#을 이용하면 주석을 표현할 수 있다.

스프링 설정을 위해 반드시 YAML을 써야 하는 것은 아니다!

application.properties를 사용할 수 있다!

Dev profile을 사용하려면?!

application-dev.properties

Lombok

getter, setter, 생성자와 같은 반복되는
보일러 플레이트 코드(boiler plate code)를 제거할 수 있다!

Lombok

- 1) lombok 의존성 추가
- 2) IntelliJ lombok 플러그인 추가
- 3) IntelliJ Annotation Processor 설정

@Getter

@NoArgsConstructor

@RequiredArgsConstructor

이 외에도...

@Setter / @EqualsAndHashCode / @ToString

Java14+ 를 사용한다면 record class를!

Kotlin을 사용한다면 언어만의 특징을 이용할 수 있다!

52강. Spring Boot 2.7.x에서 3.0.x로 업데이트하기

Spring Boot 3.0.x에서 변경된 점!

[1] Java 최소 버전이 17로 업그레이드 되었다!



[2] 많은 스프링 프로젝트, Thrid-party Library 버전업

Spring Framework 6.0.x

Hibrenate 6.1

[2] 많은 스프링 프로젝트, Thrid-party Library 버전업

Other Spring projects upgraded in this release include:

- [Spring AMQP 3.0.](#)
- [Spring Batch 5.0.](#)
- [Spring Data 2022.0.](#)
- [Spring GraphQL 1.1.](#)
- [Spring HATEOAS 2.0.](#)
- [Spring Integration 6.0.](#)
- [Spring Kafka 3.0.](#)
- [Spring LDAP 3.0.](#)
- [Spring REST Docs 3.0.](#)
- [Spring Retry 2.0.](#)
- [Spring Security 6.0](#) (see also [what's new](#)).
- [Spring Session 3.0](#)
- [Spring WS 4.0.](#)

[3] AOT 기초 작업이 이루어졌다.

AOT(Ahead of Time)

빌드를 할 때 스프링 애플리케이션을 분석하고 최적화하는 도구

[3] AOT 기초 작업이 이루어졌다.

AOT(Ahead of Time)

애플리케이션 시작 시간과 메모리 사용량을 줄일 수 있게 해준다.

[4] javax 대신 jakarta 패키지를 사용하게 된다.

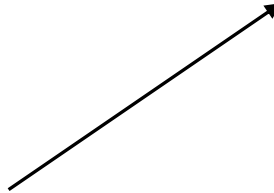


javax 패키지
(java EE)

[4] javax 대신 jakarta 패키지를 사용하게 된다.



javax 패키지
(java EE)



개발자들이
직접 사용

[4] javax 대신 jakarta 패키지를 사용하게 된다.



javax 패키지의 인기가 없어지고, 최신 기술이 반영되지 않음

[4] javax 대신 jakarta 패키지를 사용하게 된다.



ORACLE®

javax 패키지
(java EE)

오라클이 썬 마이크로시스템즈를 인수!

[4] javax 대신 jakarta 패키지를 사용하게 된다.



그리고 java EE 프로젝트를 비영리 단체인 이클립스에 이관!

[4] javax 대신 jakarta 패키지를 사용하게 된다.



그런데 "java"라는 이름의 상표권은 오라클이 가지고 있다!

[4] javax 대신 jakarta 패키지를 사용하게 된다.



그래서 jakarta로 이름을 변경!

[5] 모니터링 기능들의 강화

[6] 이 외에도 다양한 세부적인 변경사항이 많다!

자 그럼 이제 Spring Boot 3.0.x로 마이그레이션 해보자!

감사합니다