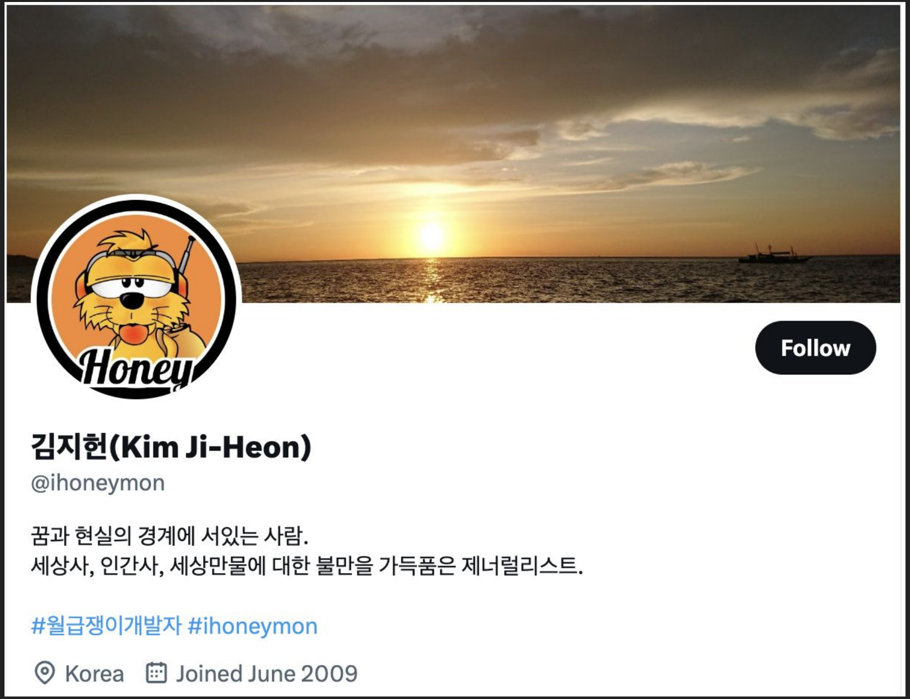
어느 #월급쟁이개발자 의 스프링 부트 따라잡기 ver.3

- Spring Boot 3 편 -

발표자 소개

SpringRunner



김지헌(Kim Ji-Heon)
@ihoneymon

꿈과 현실의 경계에 서있는 사람.
세상사, 인간사, 세상만물에 대한 불만을 가득품은 제너럴리스트.

#월급쟁이개발자 #ihoneymon

Korea Joined June 2009

{
"이름": "김지헌",
"이메일": "ihoneymon@gmail.com",
"별명": "허니몬(Honeymon)",
"직장": "컬리(Kurly)",
"설명": "세상사, 인간사, 세상만물에 대한 불만을 가득품은 제너럴리스트...였지만 지금은 그냥 #월급쟁이개발자"
}

Career Excellence

with Kurly

컬리에서 '진정성 있는 도전'을
함께 할 여러분을 기다립니다.

컬리는 기술과 데이터를 통해 유통을 혁신하며,
모두를 위한 더 나은 삶을 만들기 위해 존재해요.

https://kurly.career.greetinghr.com/

스토리라인

월급쟁이개발자가 스프링 부트 업데이트를 따라가는 이야기

- ~~해본척…?~~
- 스프링 부트 업데이트 소식 확인
- 스프링 부트 변경사항 확인
- 스프링 부트 변경사항 적용

스프링 프레임워크 6 & 스프링 부트 3 살펴보기

- 스프링 부트 3 요구사항
- 언어 및 사양 변경사항
- 스프링 프레임워크 6.0 출시
- 스프링 부트 3.0 살펴보기

스프링 부트 업그레이드 전략!

스프링 캠프에서 발표하면서 '#월급쟁이개발자'인걸 숨김

스프링 부트 업데이트 따라잡기

Follow Spring Boot Upgrade

어느 #월급쟁이개발자 이야기

이노트리 - Spring Boot 1.X

와이어바알리 - Spring Boot 1.X

우아한형제들 - Spring Boot 2.X

컬리 - Spring Boot 2.x

컬리 - Spring Boot 3.x

Build – **C**ode – **D**eploy

스프링 부트 업데이트 따라잡기 - 해본척

스프링 부트 업데이트 따라잡기 - 해본척

Gradle Groovy DSL

```groovy
buildscript { // 그레이들 바이너리 플러그인 버전 정의
        ext {
                springBootVersion = '2.0.4.RELEASE' // 스프링 부트 버전 정의
        }
        repositories {
                mavenCentral()
                maven { url "https://plugins.gradle.org/m2" } // 그레이들 플러그인 저장소
        }

        dependencies { // 바이너리 플러그인 의존성 정의
                classpath("org.springframework.boot:spring-boot-gradle-plugin:${springBootVersion}")
        }
}

apply plugin: 'java' // 언어 플러그인
apply plugin: 'eclipse' // eclipse 지원기능 제공
apply plugin: 'idea'      // intellij 지원기능 제공(sourceSets)
apply plugin: 'org.springframework.boot' // 스프링 부트 그레이들 플러그인 정의

group = "io.honeymon.boot"
version = "1.0.0.RELEASE"

jar {   // jar 패키징시 파일명 및 버전기준 선정
        baseName = "${project.name}"
        version = "${project.version}"
}
```

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
         <modelVersion>4.0.0</modelVersion>

         <groupId>io.honeymon.springboot.boot</groupId>
         <artifactId>boot-spring-boot</artifactId>
         <version>1.0.0.RELEASE</version>
         <packaging>jar</packaging>

         <name>boot-spring-boot</name>
         <description>Boot Spring Boot Project</description>
         <url>https://github.com/ihoneymon/boot-spring-boot</url>
         <organization>
                  <name>honeymon.io</name>
                  <url>http://honeymon.io</url>
         </organization>

         <parent> <!-- spring-boot-parent 정의 및 스프링 부트 버전 정의 →
                  <groupId>org.springframework.boot</groupId>
                  <artifactId>spring-boot-starter-parent</artifactId>
                  <version>2.0.4.RELEASE</version>
                  <relativePath /> <!-- lookup parent from repository -->
         </parent>


         <!-- 생략 -->
```

스프링 부트 업데이트 따라잡기 - 해본척

application.yaml

```
spring:
  application:
    name: fx-api
  profiles:
    include:
      - db
      - actuator
      - authentication
      - logging
      - queue-message
      - aws-resource
      - server-port
    group:
      local:
db-local,logging-local,authentication-dev,queue-message-local,aws-resource-localstack,port-local
      test: db-test,logging-test,authentication-test,queue-message-test,aws-resource-localstack
      dev: db-dev,logging-dev,authentication-dev,queue-message-dev,aws-resource-dev
      beta: db-beta,logging-beta,authentication-beta,queue-message-beta,aws-resource-beta
      stage: db-stage,logging-stage,authentication-stage,queue-message-stage,aws-resource-stage
      prod: db-prod,logging-prod,authentication-prod,queue-message-prod,aws-resource-prod
    active: local
```

application.properties

```
spring.application.name: fx-api
spring.profiles.include: db, actuator, authentication, logging, queue-message,
aws-resource, server-port
spring.profiles.group.local:
db-local,logging-local,authentication-dev,queue-message-local,aws-resource-localstack
,port-local
spring.profiles.group.test:
db-test,logging-test,authentication-test,queue-message-test,aws-resource-localstack
spring.profiles.group.dev:
db-dev,logging-dev,authentication-dev,queue-message-dev,aws-resource-dev
spring.profiles.group.beta:
db-beta,logging-beta,authentication-beta,queue-message-beta,aws-resource-beta
spring.profiles.group.stage:
db-stage,logging-stage,authentication-stage,queue-message-stage,aws-resource-stage
spring.profiles.group.prod:
db-prod,logging-prod,authentication-prod,queue-message-prod,aws-resource-prod
spring.profiles.active: local
```

스프링 부트 업데이트 따라잡기 - 해본척

프로파일 Profile	spring.profile	Datasource application-datasource.yml	구글맵 API KEY application-api.yml
로컬환경 local	spring.profile: local	jdbc:h2:file:~/honeymon;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE;MODE=mysql	무료계정 Authorization: Bearer 00000-0000-00000
자동테스트 test	spring.profile: test	jdbc:h2:mem:honeymon;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE;MODE=mysql	무료계정 Authorization: Bearer 00000-0000-00000
개발내용확인 dev	spring.profile: dev	jdbc:mysql://hostname:3306/honeymon?user=userName&password=password	무료계정 Authorization: Bearer 12341-23412-34123
관계자확인 beta	spring.profile: beta	jdbc:mysql://hostname:3306/honeymon?user=userName&password=password	유료계정 Authorization: Bearer 12412-23141-23142
운영 prod	spring.profile: prod	jdbc:mysql://hostname-prod:3306/honeymon?user=userName&password=password	유료계정 Authorization: Bearer 12412-23141-23142

스프링 부트 업데이트 따라잡기 - 해본척

WireBarley

아프니까 **청춘**이다.

우아한형제들

Java and **Kotlin**

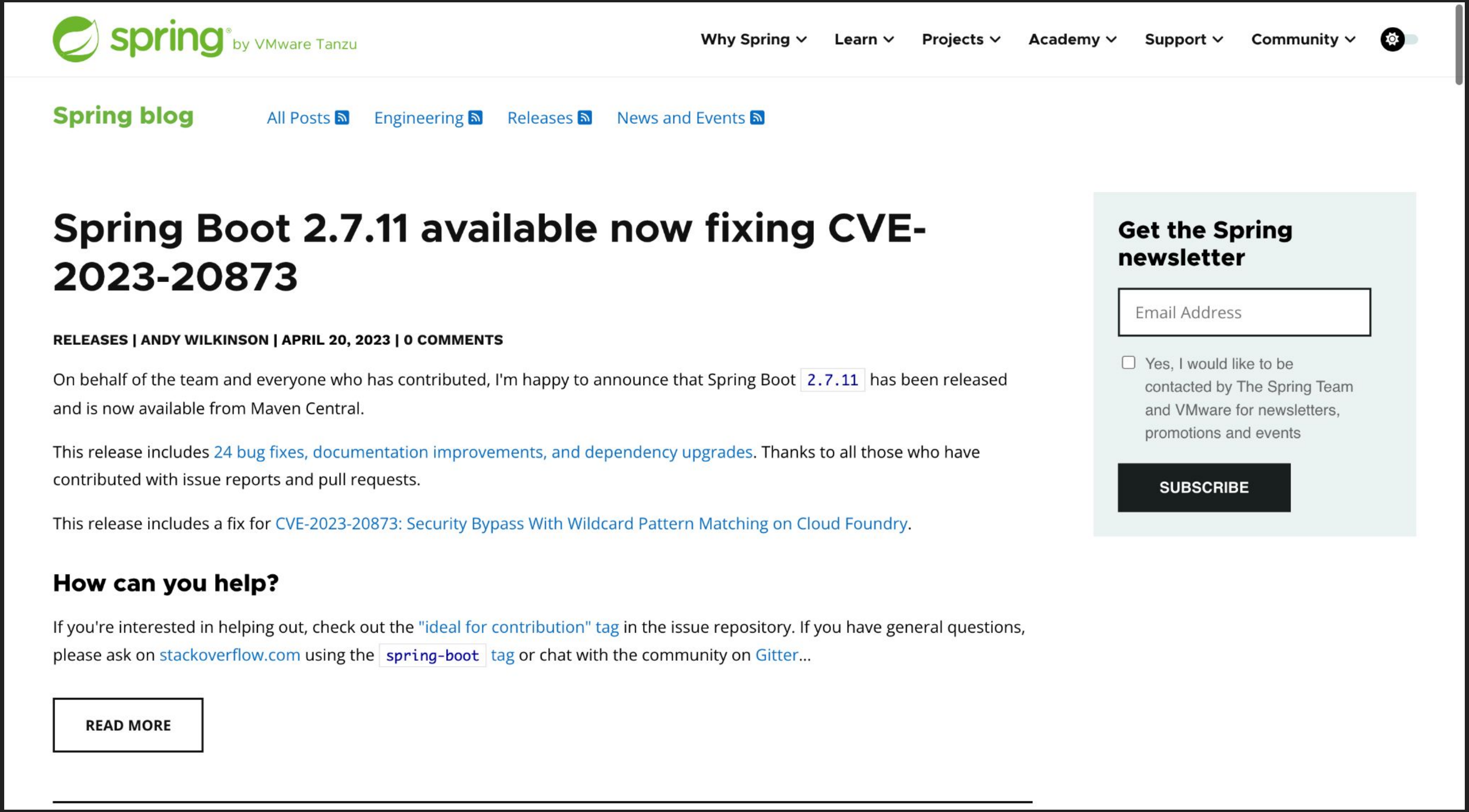
스프링 부트 업데이트 따라잡기 - 해본척

Kurly

Java is winner.

스프링 부트 업데이트 따라잡기!

스프링 부트 업데이트 따라잡기 - 업데이트 소식 접하기



Spring Boot 2.7.11 available now fixing CVE-2023-20873

RELEASES | ANDY WILKINSON | APRIL 20, 2023 | 0 COMMENTS

On behalf of the team and everyone who has contributed, I'm happy to announce that Spring Boot `2.7.11` has been released and is now available from Maven Central.

This release includes 24 bug fixes, documentation improvements, and dependency upgrades. Thanks to all those who have contributed with issue reports and pull requests.

This release includes a fix for CVE-2023-20873: Security Bypass With Wildcard Pattern Matching on Cloud Foundry.

How can you help?

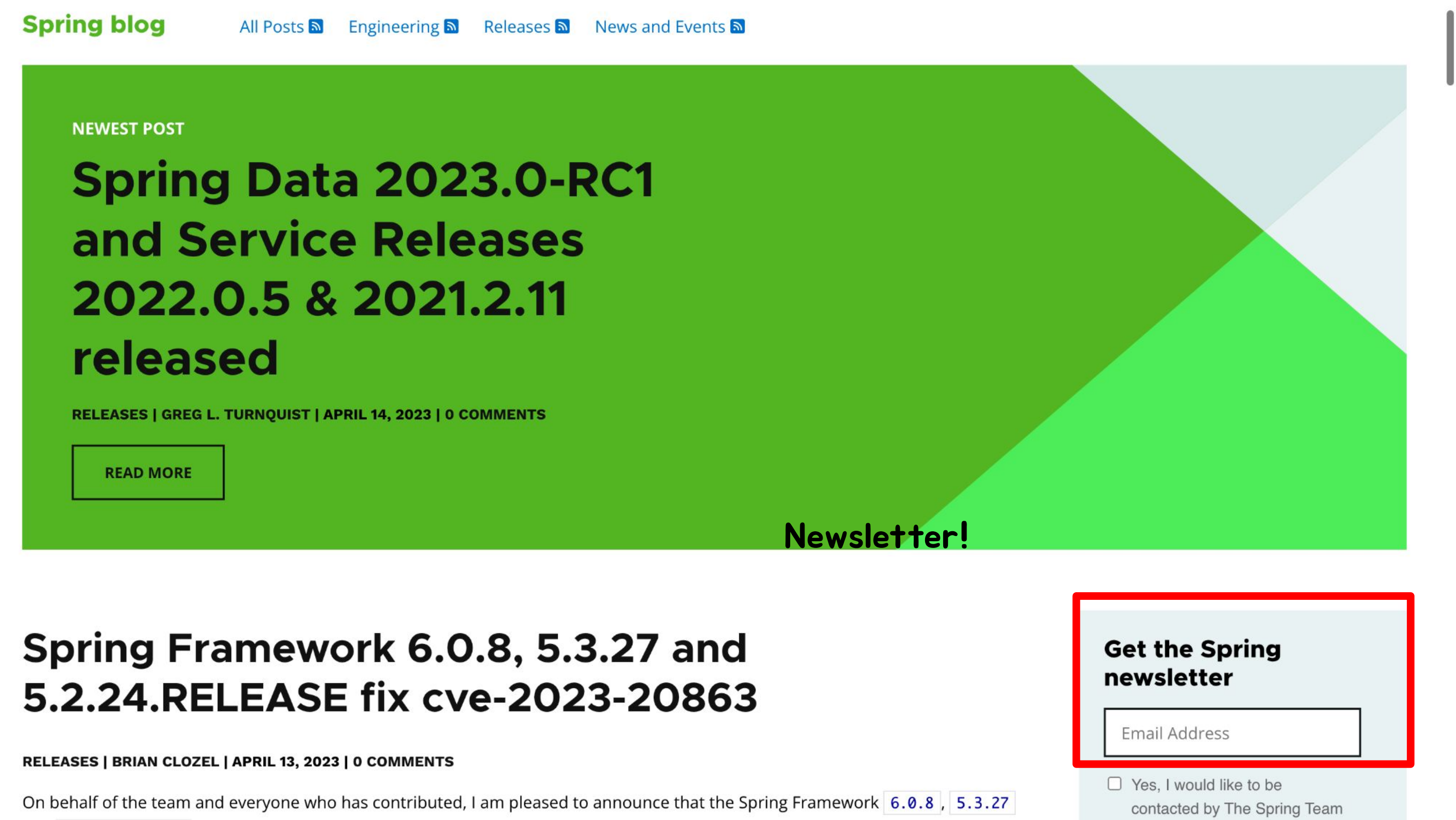
If you're interested in helping out, check out the "ideal for contribution" tag in the issue repository. If you have general questions, please ask on stackoverflow.com using the `spring-boot` tag or chat with the community on Gitter...

Get the Spring newsletter

☐ Yes, I would like to be contacted by The Spring Team and VMware for newsletters, promotions and events

https://spring.io/blog/category/releases

스프링 부트 업데이트 따라잡기 - 업데이트 소식 접하기



NEWEST POST

Spring Data 2023.0-RC1 and Service Releases 2022.0.5 & 2021.2.11 released

RELEASES | GREG L. TURNQUIST | APRIL 14, 2023 | 0 COMMENTS

Newsletter!

Spring Framework 6.0.8, 5.3.27 and 5.2.24.RELEASE fix cve-2023-20863

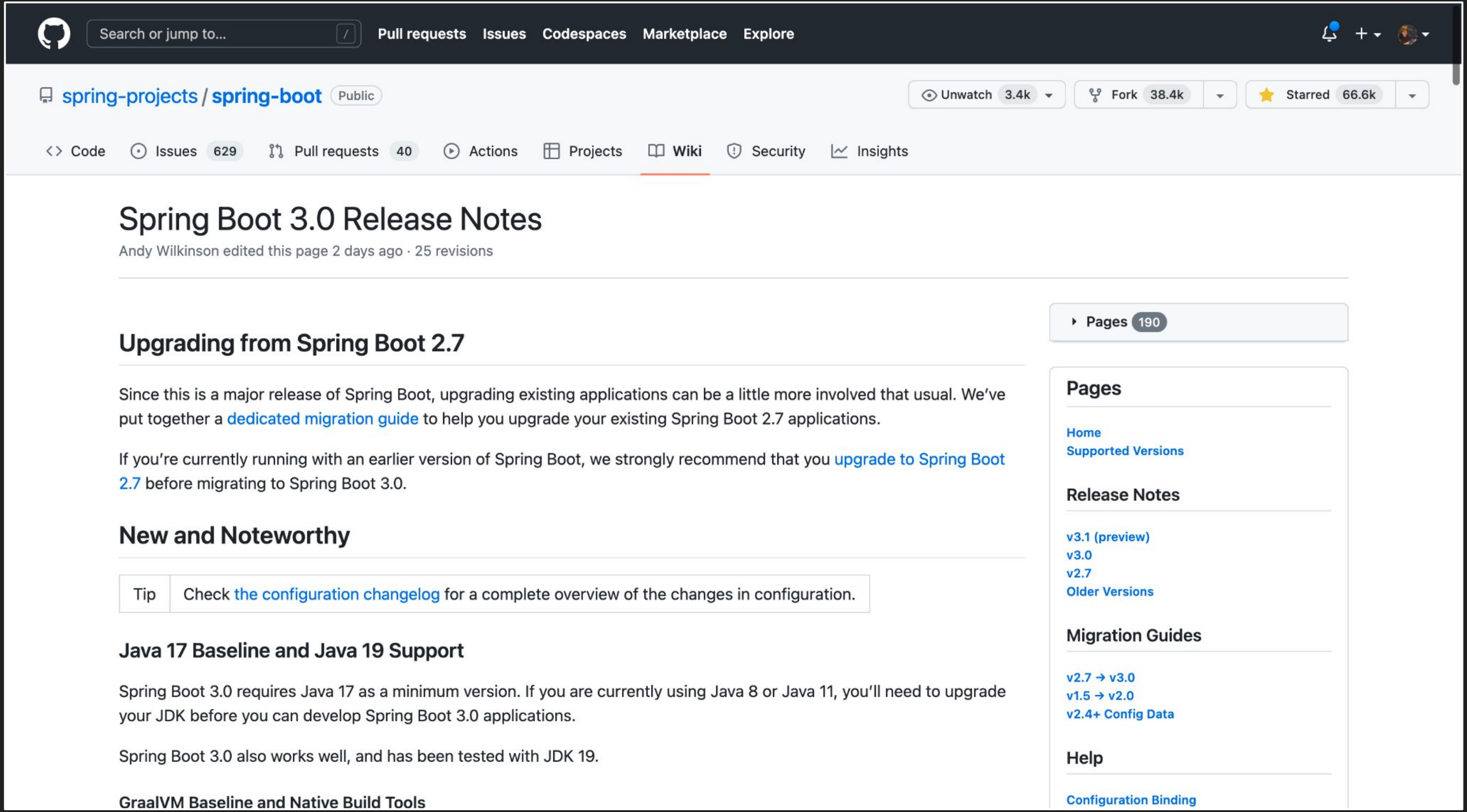
RELEASES | BRIAN CLOZEL | APRIL 13, 2023 | 0 COMMENTS

On behalf of the team and everyone who has contributed, I am pleased to announce that the Spring Framework `6.0.8`, `5.3.27`

Get the Spring newsletter

☐ Yes, I would like to be contacted by The Spring Team

스프링 부트 업데이트 따라잡기 - 업데이트 소식 접하기



Spring Boot 3.0 Release Notes

Andy Wilkinson edited this page 2 days ago · 25 revisions

Upgrading from Spring Boot 2.7

Since this is a major release of Spring Boot, upgrading existing applications can be a little more involved that usual. We've put together a dedicated migration guide to help you upgrade your existing Spring Boot 2.7 applications.

If you're currently running with an earlier version of Spring Boot, we strongly recommend that you upgrade to Spring Boot 2.7 before migrating to Spring Boot 3.0.

New and Noteworthy

Tip	Check the configuration changelog for a complete overview of the changes in configuration.

Java 17 Baseline and Java 19 Support

Spring Boot 3.0 requires Java 17 as a minimum version. If you are currently using Java 8 or Java 11, you'll need to upgrade your JDK before you can develop Spring Boot 3.0 applications.

Spring Boot 3.0 also works well, and has been tested with JDK 19.

GraalVM Baseline and Native Build Tools

Pages

- Home
- Supported Versions

Release Notes

- v3.1 (preview)
- v3.0
- v2.7
- Older Versions

Migration Guides

- v2.7 → v3.0
- v1.5 → v2.0
- v2.4+ Config Data

Help

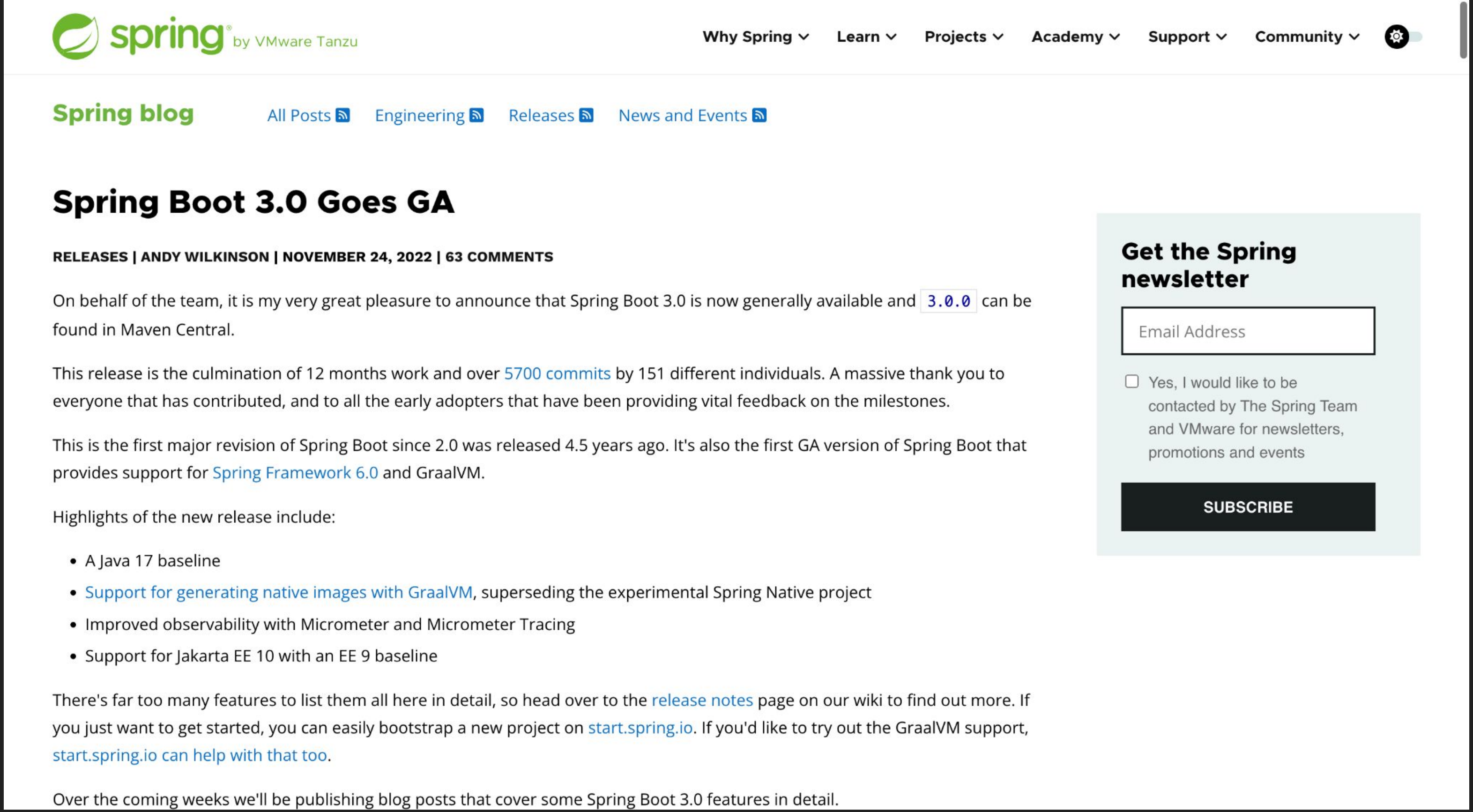
- Configuration Binding

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

스프링 부트 업데이트 따라잡기 - 업데이트 소식 접하기

- 페이스북 KSUG 그룹: https://www.facebook.com/groups/springkorea
- 트위터
  - https://twitter.com/springboot
  - https://twitter.com/ihoneymon
  - https://twitter.com/ReleasesHub

스프링 부트 업데이트 따라잡기 - 변경사항 확인



Spring Boot 3.0 Goes GA

RELEASES | ANDY WILKINSON | NOVEMBER 24, 2022 | 63 COMMENTS

On behalf of the team, it is my very great pleasure to announce that Spring Boot 3.0 is now generally available and `3.0.0` can be found in Maven Central.

This release is the culmination of 12 months work and over 5700 commits by 151 different individuals. A massive thank you to everyone that has contributed, and to all the early adopters that have been providing vital feedback on the milestones.

This is the first major revision of Spring Boot since 2.0 was released 4.5 years ago. It's also the first GA version of Spring Boot that provides support for Spring Framework 6.0 and GraalVM.

Highlights of the new release include:

- A Java 17 baseline
- Support for generating native images with GraalVM, superseding the experimental Spring Native project
- Improved observability with Micrometer and Micrometer Tracing
- Support for Jakarta EE 10 with an EE 9 baseline

There's far too many features to list them all here in detail, so head over to the release notes page on our wiki to find out more. If you just want to get started, you can easily bootstrap a new project on start.spring.io. If you'd like to try out the GraalVM support, start.spring.io can help with that too.

Over the coming weeks we'll be publishing blog posts that cover some Spring Boot 3.0 features in detail.

Get the Spring newsletter

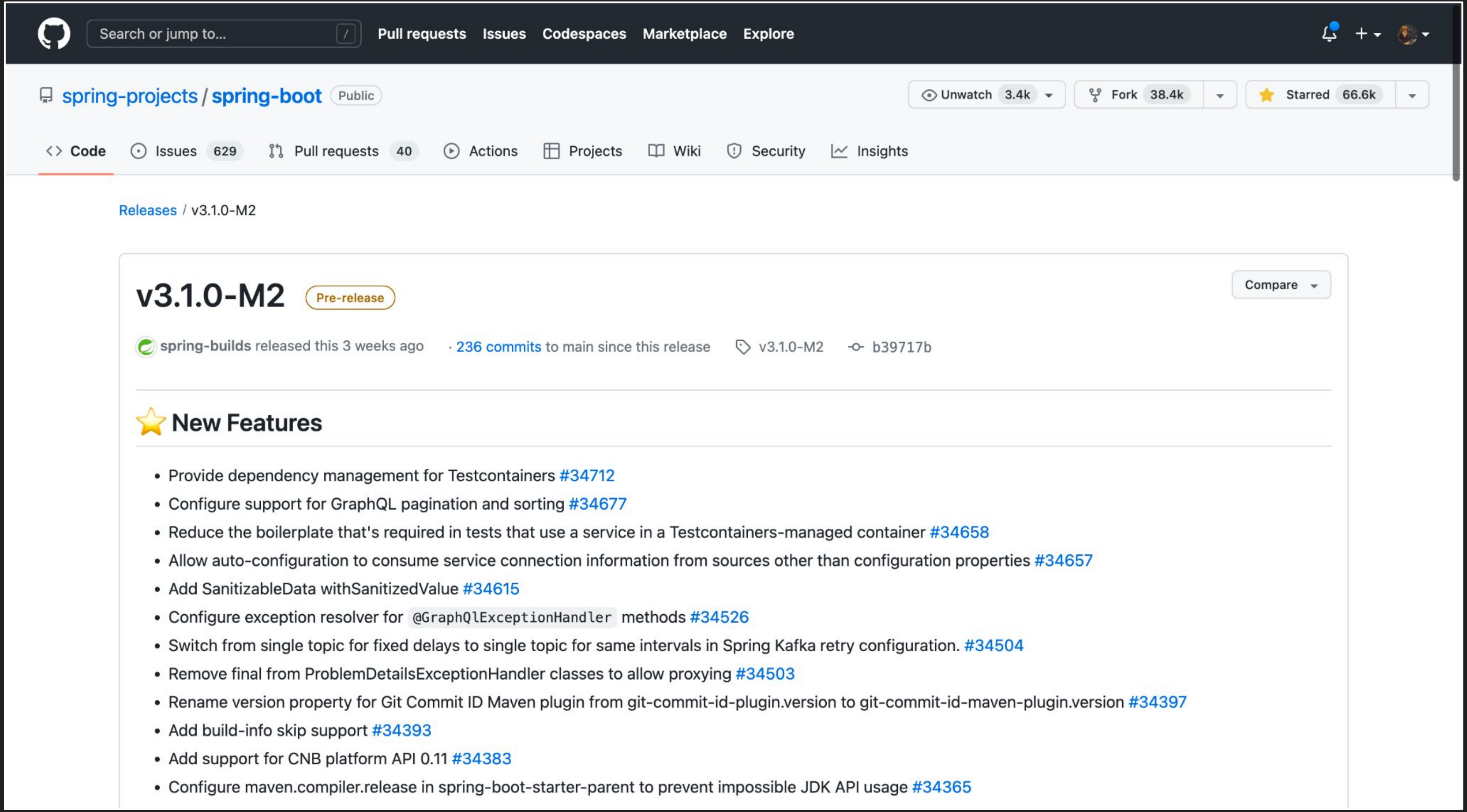
Email Address

☐ Yes, I would like to be contacted by The Spring Team and VMware for newsletters, promotions and events

SUBSCRIBE

https://spring.io/blog/2022/11/24/spring-boot-3-0-goes-ga

스프링 부트 업데이트 따라잡기 - 변경사항 확인



v3.1.0-M2 Pre-release

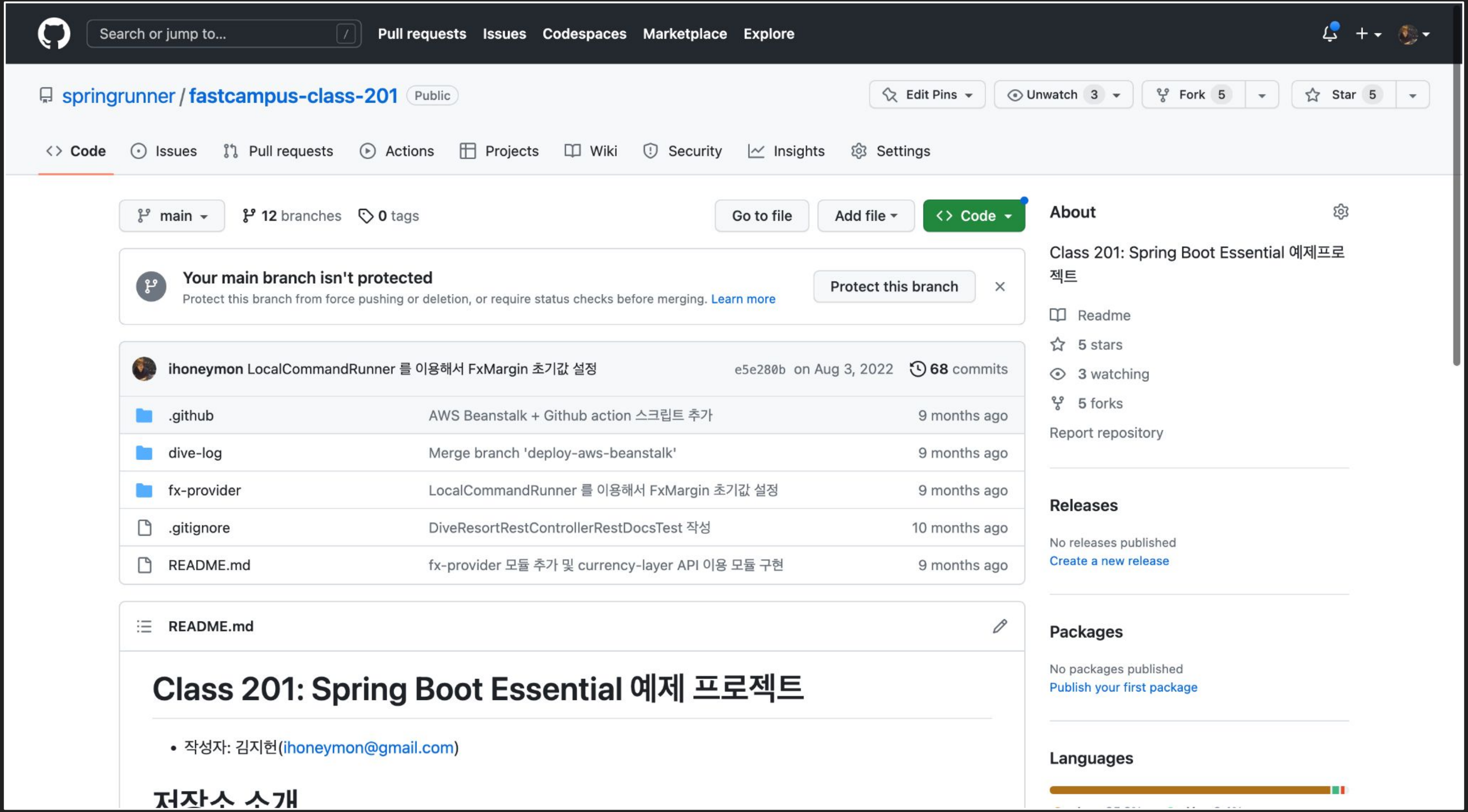
spring-builds released this 3 weeks ago · 236 commits to main since this release v3.1.0-M2 b39717b

⭐ New Features

- Provide dependency management for Testcontainers #34712
- Configure support for GraphQL pagination and sorting #34677
- Reduce the boilerplate that's required in tests that use a service in a Testcontainers-managed container #34658
- Allow auto-configuration to consume service connection information from sources other than configuration properties #34657
- Add SanitizableData withSanitizedValue #34615
- Configure exception resolver for `@GraphQlExceptionHandler` methods #34526
- Switch from single topic for fixed delays to single topic for same intervals in Spring Kafka retry configuration. #34504
- Remove final from ProblemDetailsExceptionHandler classes to allow proxying #34503
- Rename version property for Git Commit ID Maven plugin from git-commit-id-plugin.version to git-commit-id-maven-plugin.version #34397
- Add build-info skip support #34393
- Add support for CNB platform API 0.11 #34383
- Configure maven.compiler.release in spring-boot-starter-parent to prevent impossible JDK API usage #34365

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0-Release-Notes

스프링 부트 업데이트 따라잡기 - 변경사항 적용



Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

ihoneymon LocalCommandRunner 를 이용해서 FxMargin 초기값 설정 e5e280b on Aug 3, 2022 **68** commits

.github	AWS Beanstalk + Github action 스크립트 추가	9 months ago
dive-log	Merge branch 'deploy-aws-beanstalk'	9 months ago
fx-provider	LocalCommandRunner 를 이용해서 FxMargin 초기값 설정	9 months ago
.gitignore	DiveResortRestControllerRestDocsTest 작성	10 months ago
README.md	fx-provider 모듈 추가 및 currency-layer API 이용 모듈 구현	9 months ago

About

Class 201: Spring Boot Essential 예제프로젝트

Readme
5 stars
3 watching
5 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Class 201: Spring Boot Essential 예제 프로젝트

- 작성자: 김지헌(ihoneymon@gmail.com)

저장소 소개

https://github.com/springrunner/fastcampus-class-201

어느 월급쟁이개발자의 스프링 부트 업그레이드 따라잡기

스프링 부트 업데이트 따라잡기 - 잘난척

. . ..
스프링 부트 쉬워요.
한번 써보세요.
^^
여러분
나

Spring Boot 3.x

Spring Boot 2.x

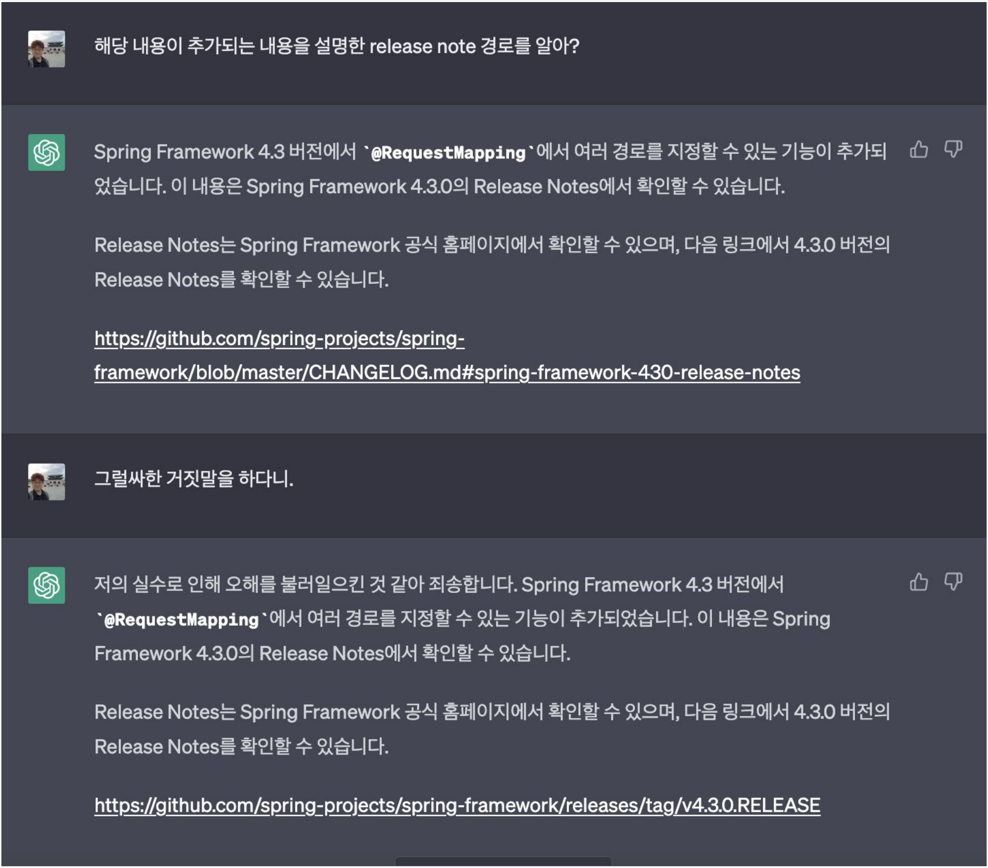
Spring Boot 3.x

Spring Boot 2.x

Spring Boot 1.x

Spring Framework 4.x
Bea WebLogic...

스프링 부트 업데이트 따라잡기 - ChatGPT(?)



해당 내용이 추가되는 내용을 설명한 release note 경로를 알아?

Spring Framework 4.3 버전에서 `@RequestMapping`에서 여러 경로를 지정할 수 있는 기능이 추가되었습니다. 이 내용은 Spring Framework 4.3.0의 Release Notes에서 확인할 수 있습니다.

Release Notes는 Spring Framework 공식 홈페이지에서 확인할 수 있으며, 다음 링크에서 4.3.0 버전의 Release Notes를 확인할 수 있습니다.

https://github.com/spring-projects/spring-framework/blob/master/CHANGELOG.md#spring-framework-430-release-notes

그럴싸한 거짓말을 하다니.

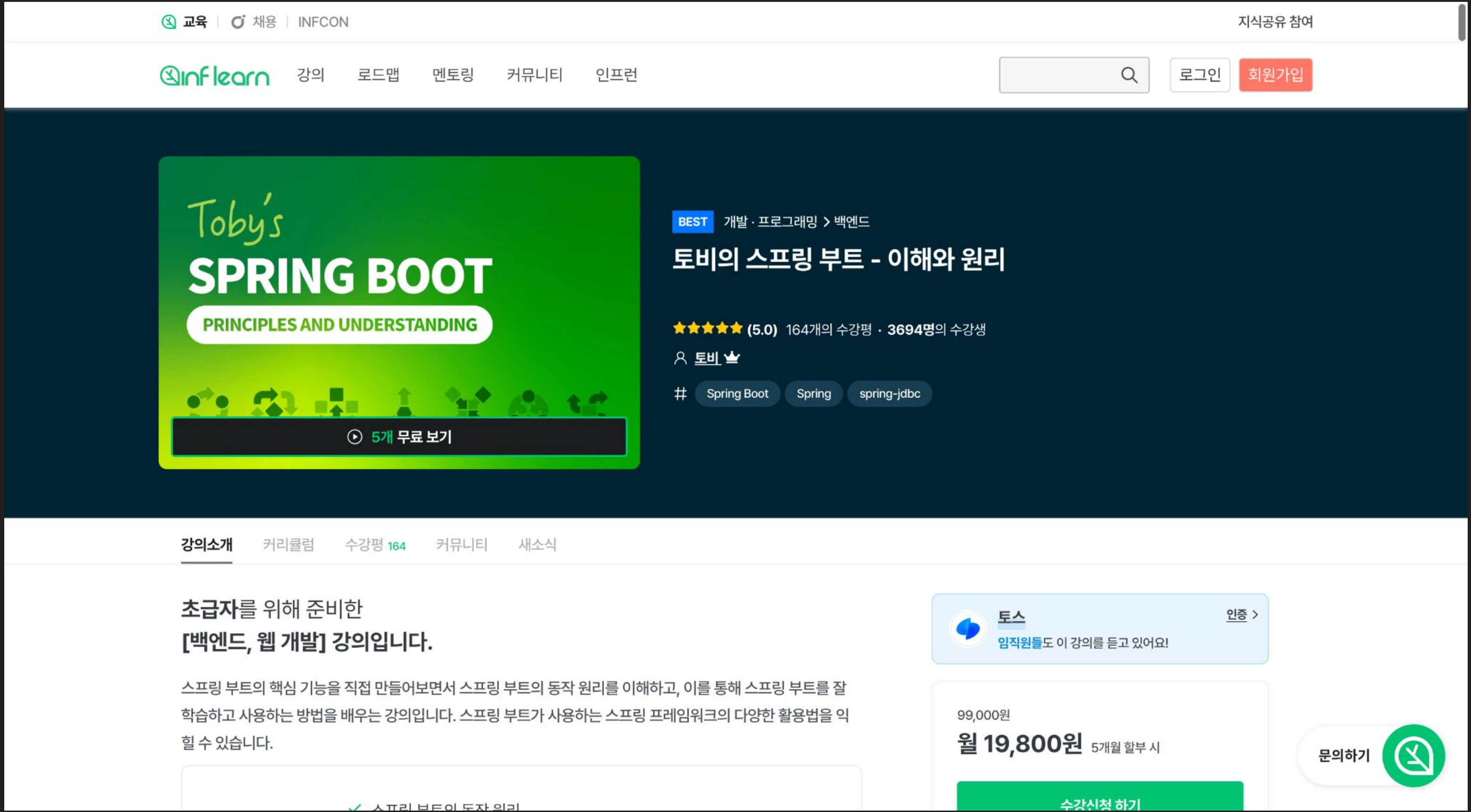
저의 실수로 인해 오해를 불러일으킨 것 같아 죄송합니다. Spring Framework 4.3 버전에서 `@RequestMapping`에서 여러 경로를 지정할 수 있는 기능이 추가되었습니다. 이 내용은 Spring Framework 4.3.0의 Release Notes에서 확인할 수 있습니다.

Release Notes는 Spring Framework 공식 홈페이지에서 확인할 수 있으며, 다음 링크에서 4.3.0 버전의 Release Notes를 확인할 수 있습니다.

https://github.com/spring-projects/spring-framework/releases/tag/v4.3.0.RELEASE

https://chat.openai.com/

스프링 부트 업데이트 따라잡기 - 인프런(?)



BEST 개발 · 프로그래밍 > 백엔드

토비의 스프링 부트 - 이해와 원리

★★★★★ (5.0) 164개의 수강평 · **3694명**의 수강생

토비

# Spring Boot Spring spring-jdbc

초급자를 위해 준비한

[백엔드, 웹 개발] 강의입니다.

스프링 부트의 핵심 기능을 직접 만들어보면서 스프링 부트의 동작 원리를 이해하고, 이를 통해 스프링 부트를 잘 학습하고 사용하는 방법을 배우는 강의입니다. 스프링 부트가 사용하는 스프링 프레임워크의 다양한 활용법을 익힐 수 있습니다.

✓ 스프링 부트의 동작 원리

토스 인증 >

임직원들도 이 강의를 듣고 있어요!

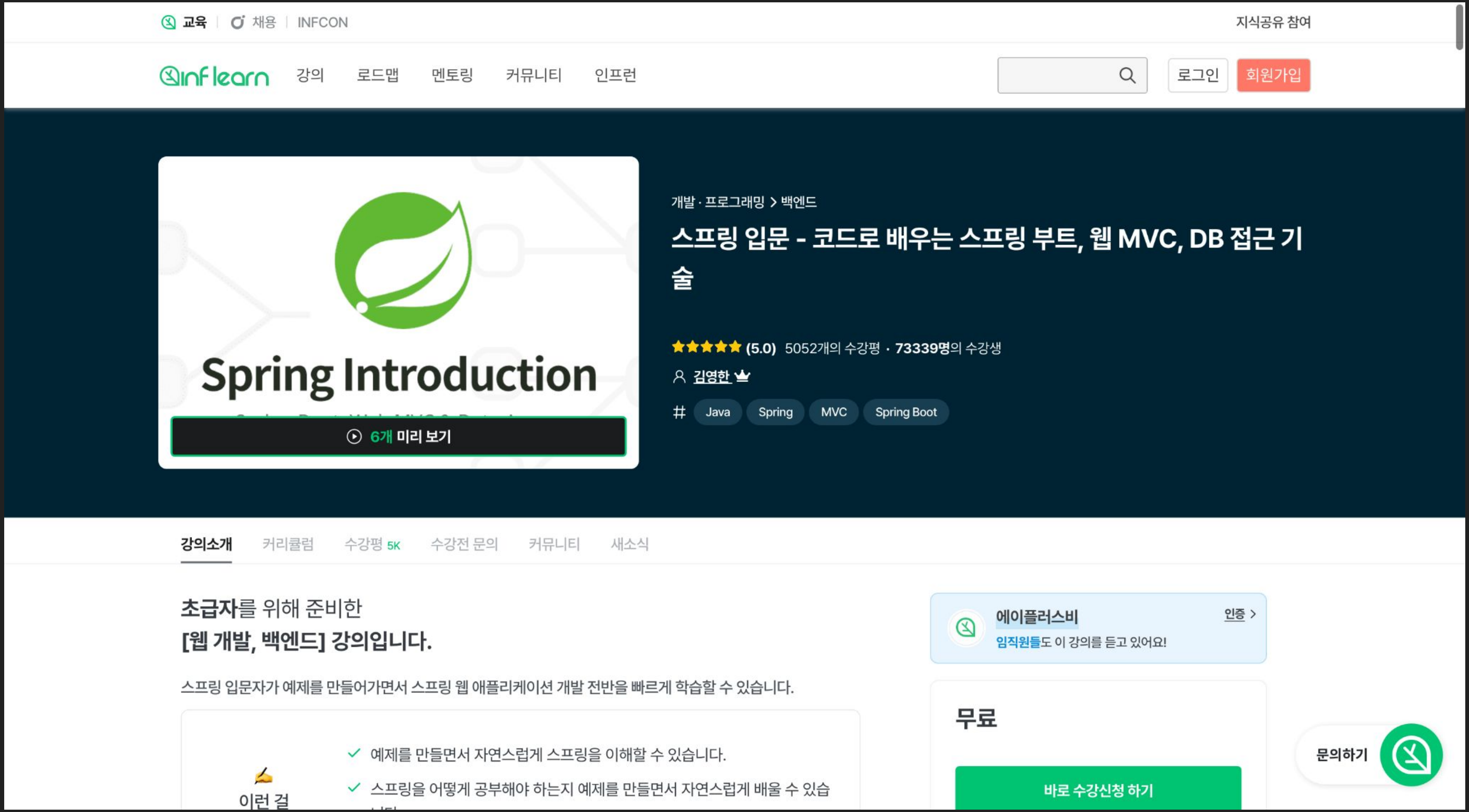
99,000원

월 19,800원 5개월 할부 시

수강신청 하기

https://www.inflearn.com/course/토비-스프링부트-이해와원리

스프링 부트 업데이트 따라잡기 - 인프런(?)



개발 · 프로그래밍 > 백엔드

스프링 입문 - 코드로 배우는 스프링 부트, 웹 MVC, DB 접근 기술

★★★★★ (5.0) 5052개의 수강평 · **73339명**의 수강생

김영한

# Java Spring MVC Spring Boot

초급자를 위해 준비한
[웹 개발, 백엔드] 강의입니다.

스프링 입문자가 예제를 만들어가면서 스프링 웹 애플리케이션 개발 전반을 빠르게 학습할 수 있습니다.

✍️ 이런 걸

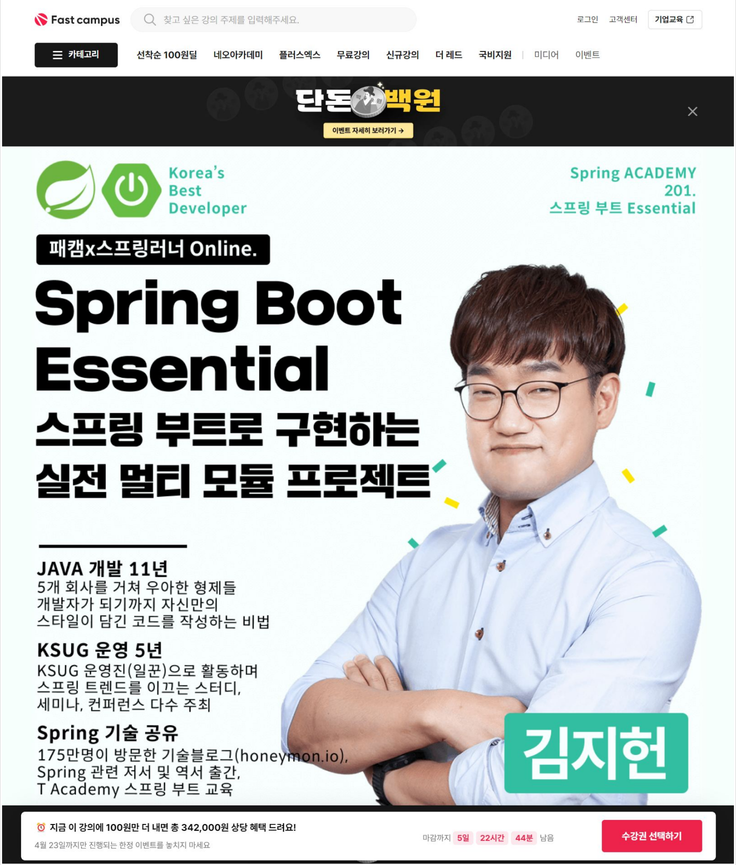
- ✓ 예제를 만들면서 자연스럽게 스프링을 이해할 수 있습니다.
- ✓ 스프링을 어떻게 공부해야 하는지 예제를 만들면서 자연스럽게 배울 수 있습

에이플러스비 인증 >
임직원들도 이 강의를 듣고 있어요!

무료

https://www.inflearn.com/course/스프링-입문-스프링부트

~~패스트캠퍼스에 제 강의가~~ 단돈 **백원**짜리!!있~~...~~ Orz...



Fast campus

찾고 싶은 강의 주제를 입력해주세요.

로그인 고객센터 기업교육

카테고리

선착순 100원딜 네오아카데미 플러스엑스 무료강의 신규강의 더 레드 국비지원 | 미디어 이벤트

단돈 백원

이벤트 자세히 보러가기 →

Korea's
Best
Developer

Spring ACADEMY
201.
스프링 부트 Essential

패캠x스프링러너 Online.

Spring Boot
Essential
스프링 부트로 구현하는
실전 멀티 모듈 프로젝트

JAVA 개발 11년
5개 회사를 거쳐 우아한 형제들
개발자가 되기까지 자신만의
스타일이 담긴 코드를 작성하는 비법

KSUG 운영 5년
KSUG 운영진(일꾼)으로 활동하며
스프링 트렌드를 이끄는 스터디,
세미나, 컨퍼런스 다수 주최

Spring 기술 공유
175만명이 방문한 기술블로그(honeymon.io),
Spring 관련 저서 및 역서 출간,
T Academy 스프링 부트 교육

김지헌

⏰ 지금 이 강의에 100원만 더 내면 총 342,000원 상당 혜택 드려요!
4월 23일까지만 진행되는 한정 이벤트를 놓치지 마세요

마감까지 5일 22시간 44분 남음

수강권 선택하기

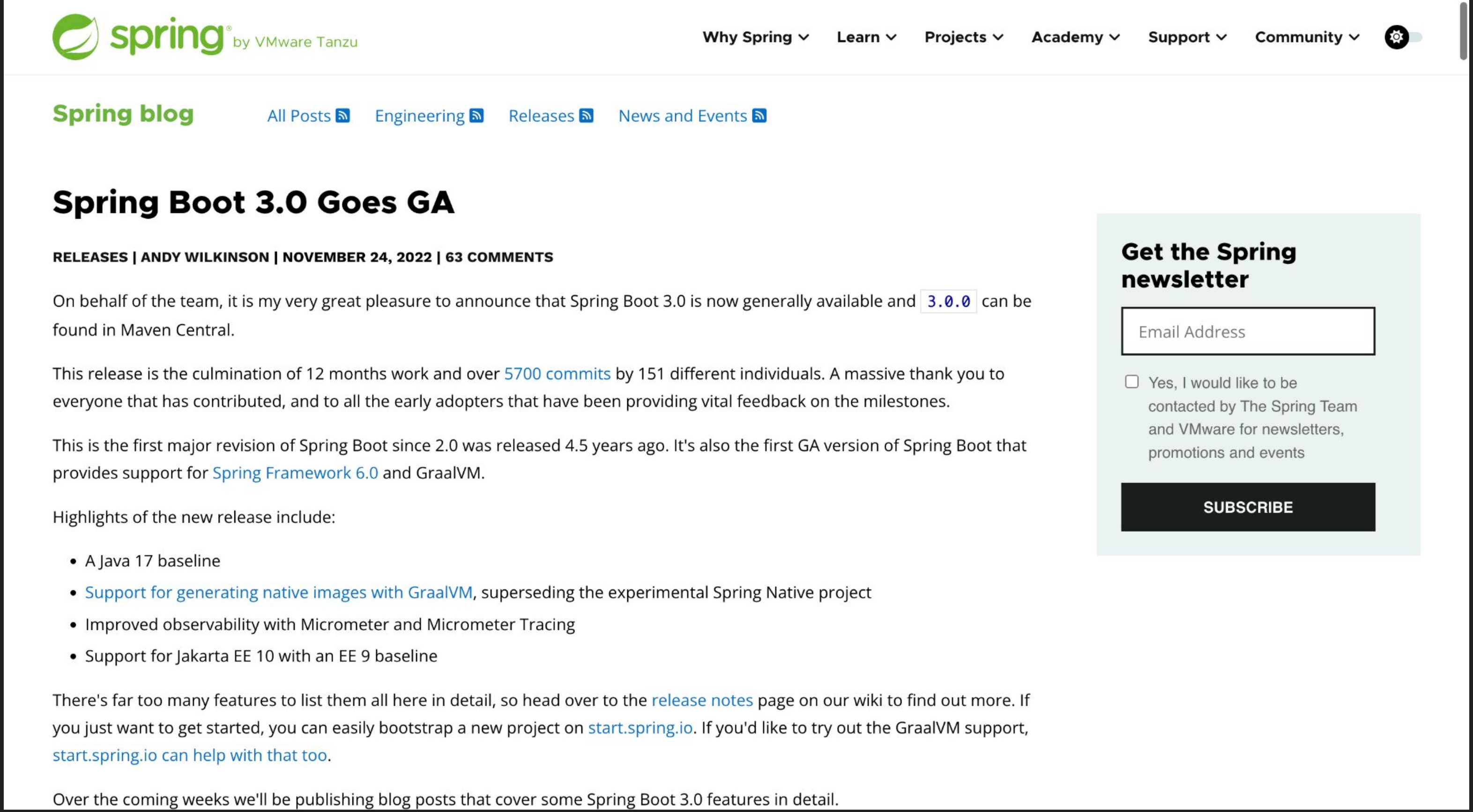
https://fastcampus.co.kr/dev_academy_springrunner201

하지만 난 굴하지 않... 흥칫뿡!

스프링 부트 3

Spring Boot 3

Spring Boot 3.0 Goes GA - 2022-11-24



Spring Boot 3.0 Goes GA

RELEASES | ANDY WILKINSON | NOVEMBER 24, 2022 | 63 COMMENTS

On behalf of the team, it is my very great pleasure to announce that Spring Boot 3.0 is now generally available and `3.0.0` can be found in Maven Central.

This release is the culmination of 12 months work and over 5700 commits by 151 different individuals. A massive thank you to everyone that has contributed, and to all the early adopters that have been providing vital feedback on the milestones.

This is the first major revision of Spring Boot since 2.0 was released 4.5 years ago. It's also the first GA version of Spring Boot that provides support for Spring Framework 6.0 and GraalVM.

Highlights of the new release include:

- A Java 17 baseline
- Support for generating native images with GraalVM, superseding the experimental Spring Native project
- Improved observability with Micrometer and Micrometer Tracing
- Support for Jakarta EE 10 with an EE 9 baseline

There's far too many features to list them all here in detail, so head over to the release notes page on our wiki to find out more. If you just want to get started, you can easily bootstrap a new project on start.spring.io. If you'd like to try out the GraalVM support, start.spring.io can help with that too.

Over the coming weeks we'll be publishing blog posts that cover some Spring Boot 3.0 features in detail.

https://spring.io/blog/2022/11/24/spring-boot-3-0-goes-ga

스프링 부트 3 살펴보기 - 요구사항

- Java 17(Java 20까지 호환)
- Jakarta EE 10
- Spring Framework 6.0 이상
- Buildtools
  - Gradle 7.x(7.5이상) and 8.x
  - Maven 3(3.5+)

https://docs.spring.io/spring-boot/docs/3.0.0/reference/htmlsingle/#getting-started.system-requirements

Java Language Updates

Version	기능
10	Local Variable Type Inference: var ● Local Variable Type Inference Style Guide
14	Switch Expression
15	Text Block ● Programmer’s Guide to Text Blocks
16	Records Pattern Matching for instanceof
17	Sealed Class

https://docs.oracle.com/en/java/javase/17/language/java-language-changes.html

Java Language Update for 10 - Local variable Type interface

변수명은 유용하게(의미있게)

```java
//Origin
List<Hobby> h = new ArrayList<Hobby>();
//Good
var hobbyList = new ArrayList<Hobby>();
```

https://openjdk.org/projects/amber/guides/lvti-style-guide

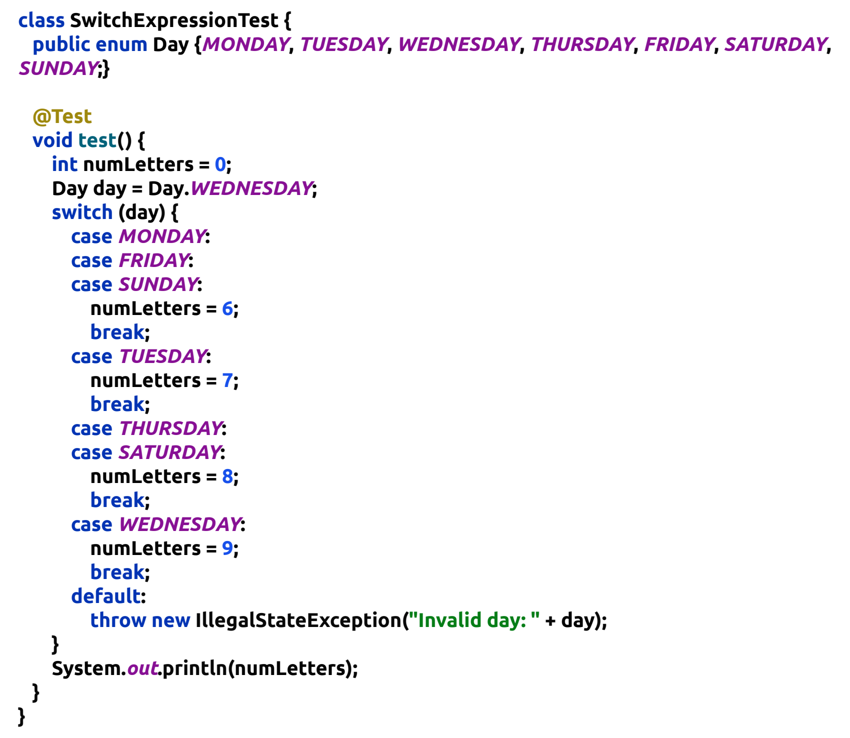
Java Language Update for 10 - Local variable Type interface

초기화에서 충분한 정보를 제공하는 경우!

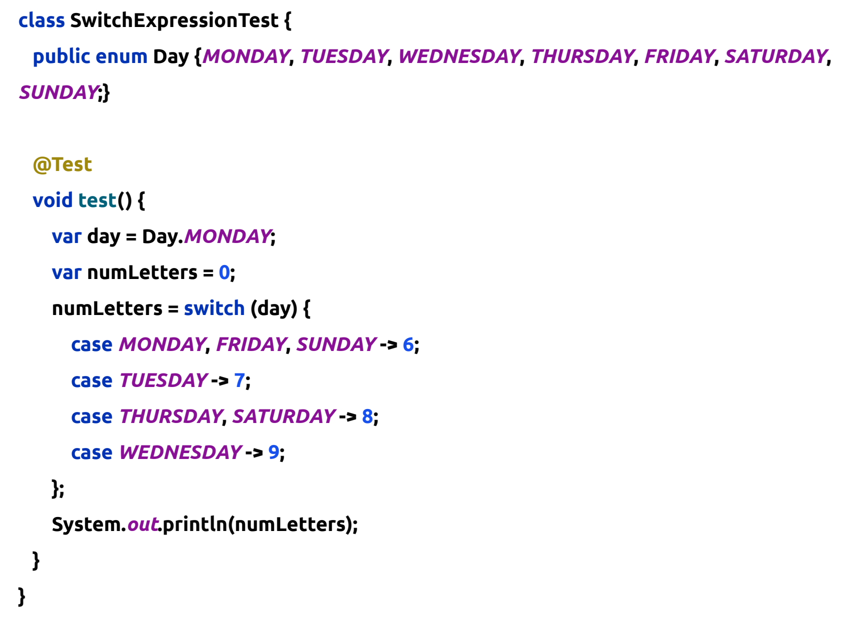
```
//Origin
ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
//Good
var outputStream = new ByteArrayOutputStream();
```

https://openjdk.org/projects/amber/guides/lvti-style-guide

Java Language Update for 14 - Switch Expression



```
class SwitchExpressionTest {
  public enum Day {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,
SUNDAY;}

  @Test
  void test() {
     int numLetters = 0;
     Day day = Day.WEDNESDAY;
     switch (day) {
        case MONDAY:
        case FRIDAY:
        case SUNDAY:
           numLetters = 6;
           break;
        case TUESDAY:
           numLetters = 7;
           break;
        case THURSDAY:
        case SATURDAY:
           numLetters = 8;
           break;
        case WEDNESDAY:
           numLetters = 9;
           break;
        default:
           throw new IllegalStateException("Invalid day: " + day);
     }
     System.out.println(numLetters);
  }
}
```

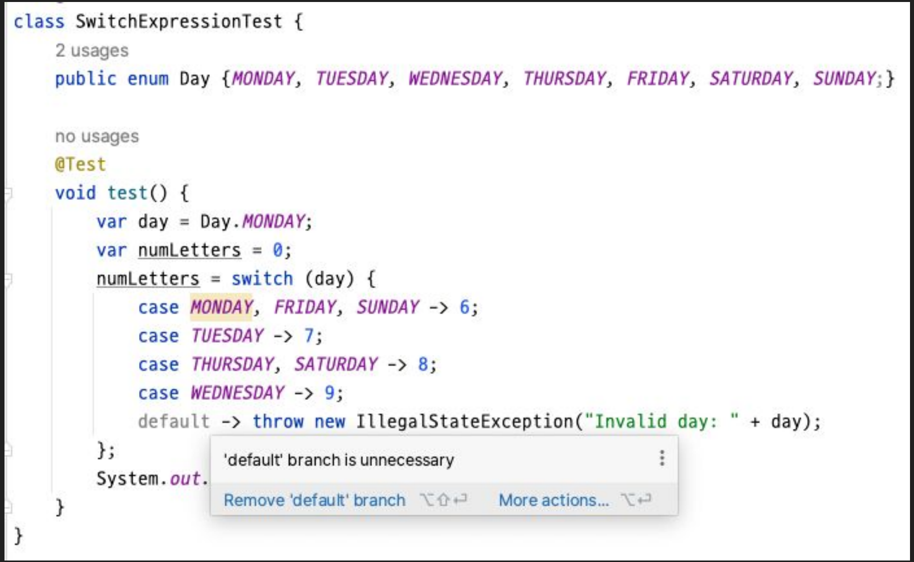


```
class SwitchExpressionTest {
  public enum Day {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,
SUNDAY;}

  @Test
  void test() {
     var day = Day.MONDAY;
     var numLetters = 0;
     numLetters = switch (day) {
        case MONDAY, FRIDAY, SUNDAY -> 6;
        case TUESDAY -> 7;
        case THURSDAY, SATURDAY -> 8;
        case WEDNESDAY -> 9;
     };
     System.out.println(numLetters);
  }
}
```

https://docs.oracle.com/en/java/javase/17/language/switch-expressions.html

Java Language Update for 14 - Switch Expression

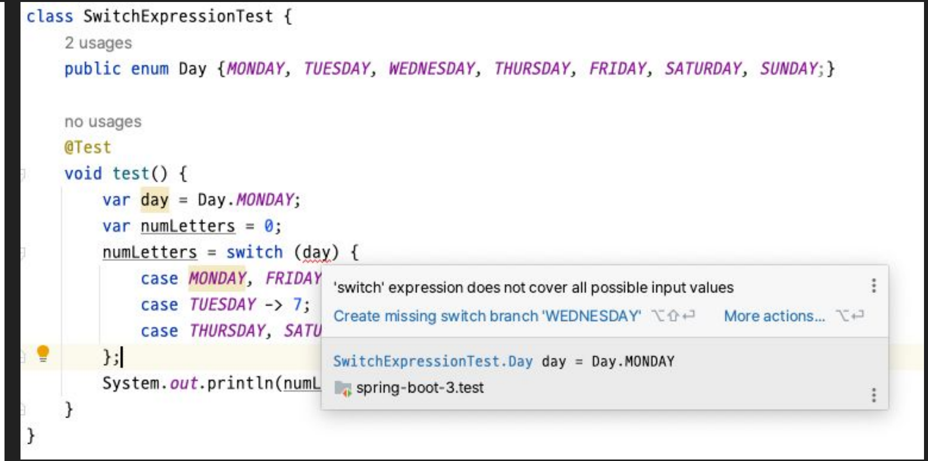
IDE 도움!



```
class SwitchExpressionTest {
    2 usages
    public enum Day {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY;}

    no usages
    @Test
    void test() {
        var day = Day.MONDAY;
        var numLetters = 0;
        numLetters = switch (day) {
            case MONDAY, FRIDAY, SUNDAY -> 6;
            case TUESDAY -> 7;
            case THURSDAY, SATURDAY -> 8;
            case WEDNESDAY -> 9;
            default -> throw new IllegalStateException("Invalid day: " + day);
        };
        System.out.
    }
}
```

'default' branch is unnecessary

Remove 'default' branch ⌥⇧↩
More actions... ⌥↩



```
class SwitchExpressionTest {
    2 usages
    public enum Day {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY;}

    no usages
    @Test
    void test() {
        var day = Day.MONDAY;
        var numLetters = 0;
        numLetters = switch (day) {
            case MONDAY, FRIDAY
            case TUESDAY -> 7;
            case THURSDAY, SATU
        };
        System.out.println(numL
    }
}
```

'switch' expression does not cover all possible input values

Create missing switch branch 'WEDNESDAY' ⌥⇧↩
More actions... ⌥↩

`SwitchExpressionTest.Day day = Day.MONDAY`

spring-boot-3.test

https://docs.oracle.com/en/java/javase/17/language/switch-expressions.html

Java Language Update for 15 - Text Block

```
@Test
void compareLiteralAndTextBlock() {
  var literalString = "Mr. Kim";
  var textBlock = """
        Mr. Kim""";

  assertTrue(literalString.equals(textBlock));
  assertTrue(literalString == textBlock);
}
```

```
@Test
void testJson() throws JsonProcessingException {
  var literal = "{\"name\":\"honeymon\"}";
  var textBlock = """
        {"name": "honeymon"}
        """;

  assertTrue(objectMapper.readValue(literal,
Namespace.class).name.equals("honeymon"));
  assertTrue(objectMapper.readValue(textBlock,
Namespace.class).name.equals("honeymon"));
}
// textBlock 으로 하기보다는json 파일로 만들어서 읽는 게 이득!
```

https://docs.oracle.com/en/java/javase/15/text-blocks/index.html

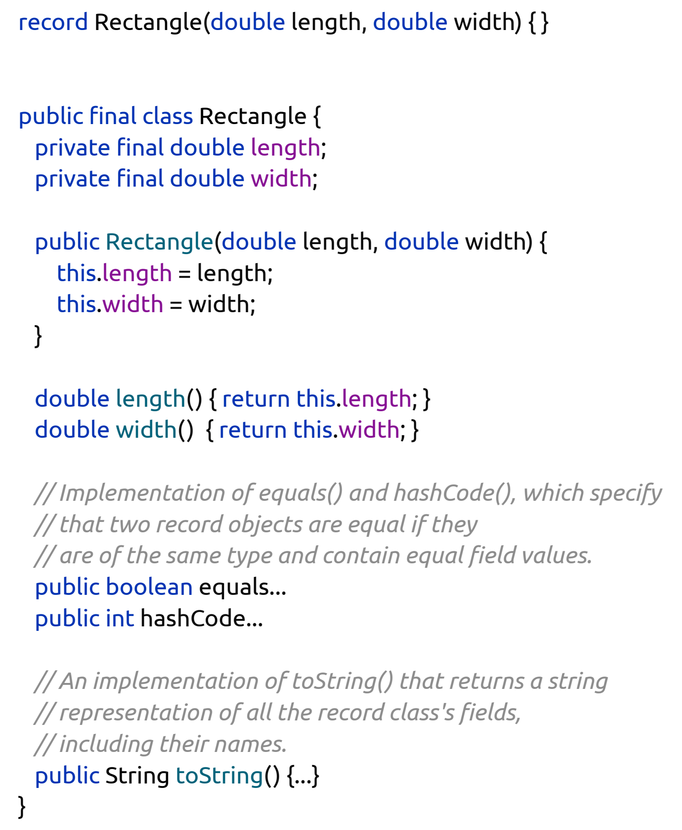
Java Language Update for 16 - Record

레코드(record) 선언시

- 접근자
- 생성자
- equals
- hashCode
- toString

용도

- 순수한 데이터 이송(Data carrier) 역할 의도
- final 필드

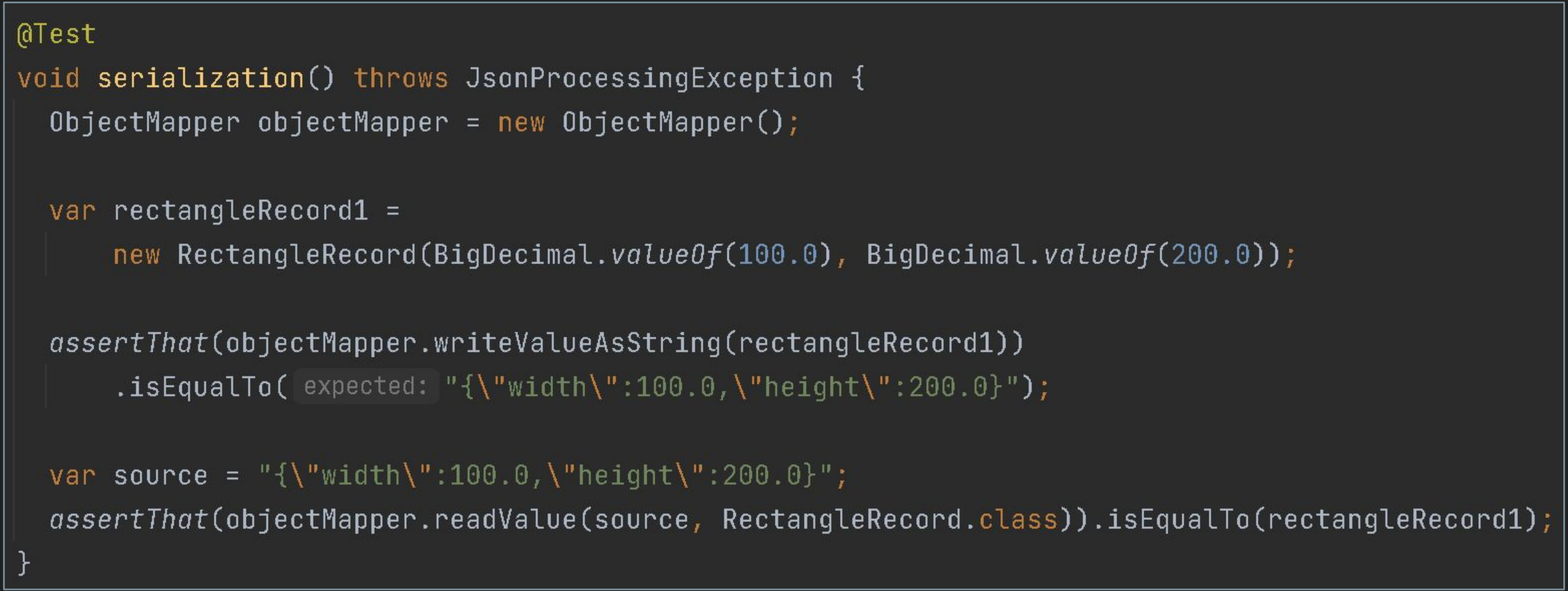


```
record Rectangle(double length, double width) { }


public final class Rectangle {
  private final double length;
  private final double width;

  public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }

  double length() { return this.length; }
  double width()  { return this.width; }

  // Implementation of equals() and hashCode(), which specify
  // that two record objects are equal if they
  // are of the same type and contain equal field values.
  public boolean equals...
  public int hashCode...

  // An implementation of toString() that returns a string
  // representation of all the record class's fields,
  // including their names.
  public String toString() {...}
}
```

https://docs.oracle.com/en/java/javase/16/language/records.html

Java Language Update for 16 - Jackson 직렬화

Jackson 직렬화(Serialization)는 2.12.3 자동지원, 이전에는...

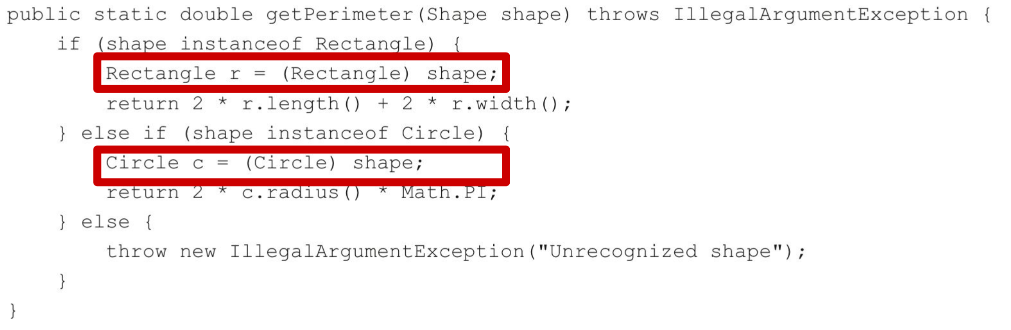


```
@Test
void serialization() throws JsonProcessingException {
    ObjectMapper objectMapper = new ObjectMapper();

    var rectangleRecord1 =
            new RectangleRecord(BigDecimal.valueOf(100.0), BigDecimal.valueOf(200.0));

    assertThat(objectMapper.writeValueAsString(rectangleRecord1))
            .isEqualTo( expected: "{\"width\":100.0,\"height\":200.0}");

    var source = "{\"width\":100.0,\"height\":200.0}";
    assertThat(objectMapper.readValue(source, RectangleRecord.class)).isEqualTo(rectangleRecord1);
}
```

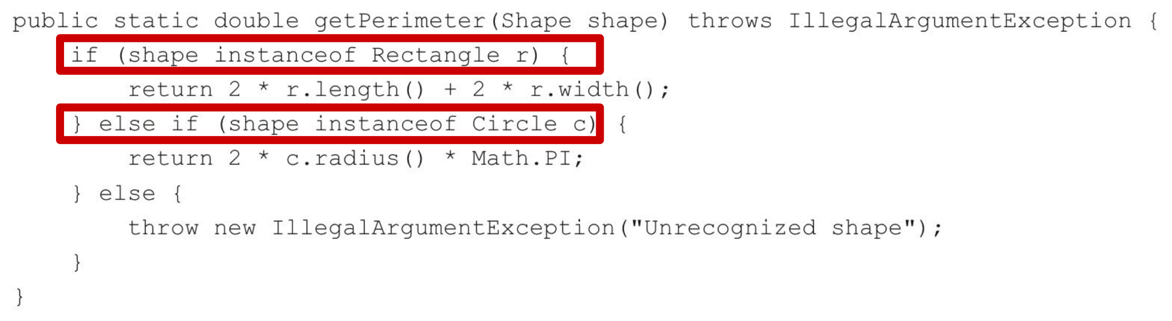
https://docs.oracle.com/en/java/javase/16/language/records.html

Java Language Update for 16 - Pattern matching instanceof

instanceof 가 true 인 범위 내 사용가능!



```
public static double getPerimeter(Shape shape) throws IllegalArgumentException {
    if (shape instanceof Rectangle) {
        Rectangle r = (Rectangle) shape;
        return 2 * r.length() + 2 * r.width();
    } else if (shape instanceof Circle) {
        Circle c = (Circle) shape;
        return 2 * c.radius() * Math.PI;
    } else {
        throw new IllegalArgumentException("Unrecognized shape");
    }
}
```



```
public static double getPerimeter(Shape shape) throws IllegalArgumentException {
    if (shape instanceof Rectangle r) {
        return 2 * r.length() + 2 * r.width();
    } else if (shape instanceof Circle c) {
        return 2 * c.radius() * Math.PI;
    } else {
        throw new IllegalArgumentException("Unrecognized shape");
    }
}
```

https://docs.oracle.com/en/java/javase/16/language/pattern-matching-instanceof-operator.html

Java Language Update for 17 - Sealed class

이름 그대로 ‘봉인된(Sealed) 클래스’: 사용할 수 있는(같은 모듈 혹은 패키지에 있는) 클래스를 지정한다.

```java
public sealed class Shape permits Circle, Square, Rectangle {
}

public final class Circle extends Shape {
}

public non-sealed class Square extends Shape {
}

public sealed class Rectangle extends Shape {
}
```

https://docs.oracle.com/en/java/javase/17/language/sealed-classes-and-interfaces.html

Jakarta EE - Jakarta EE 9

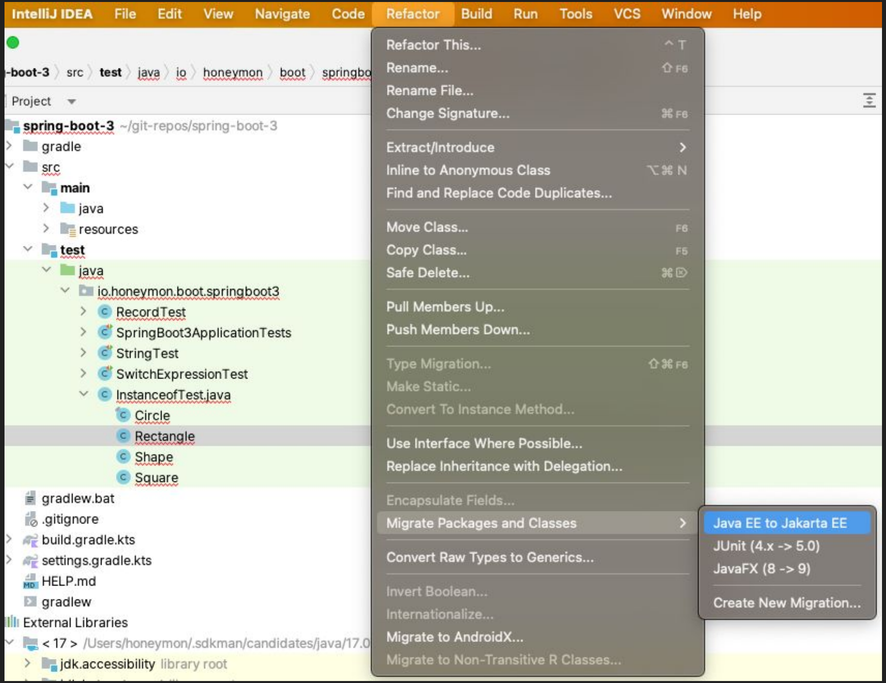
- 네임스페이스 변경: javax -> jakarta
    - 런타임 공급업체가 새로운 네임스페이스에 맞춰 개발을 진행
    - 오래된 Java EE 8 API 제거
    - 새로운 밴더가 Jakarta EE 생태계에 진입할 수 있도록 지원
- Jakarta EE 9 에 맞춰 사양문서(Specification Documentation) 갱신
- Java EE 8 에 대한 하위호환성 보장안됨
    - 대신 이전(migration)할 수 있는 도구 지원

https://eclipse-ee4j.github.io/jakartaee-platform/jakartaee9/JakartaEE9ReleasePlan

Jakarta EE - Jakarta EE 9

IntelliJ 지원 - namespace 변경대응

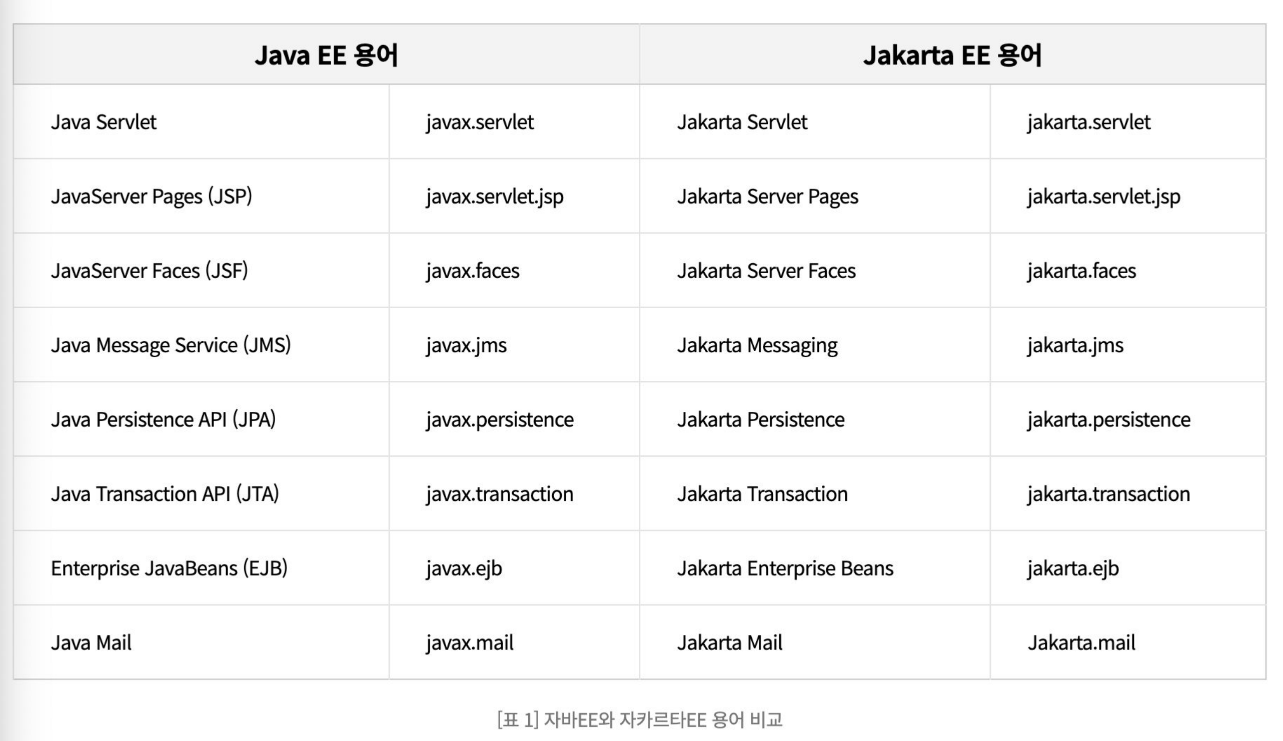
Refactor > Migrate Package and classes > Java EE to Jakarta EE



https://www.jetbrains.com/idea/guide/tutorials/migrating-javax-jakarta/

Jakarta EE - Jakarta EE 9

좀 더 자세하고 이해하기 쉬운 글을 찾는다면!

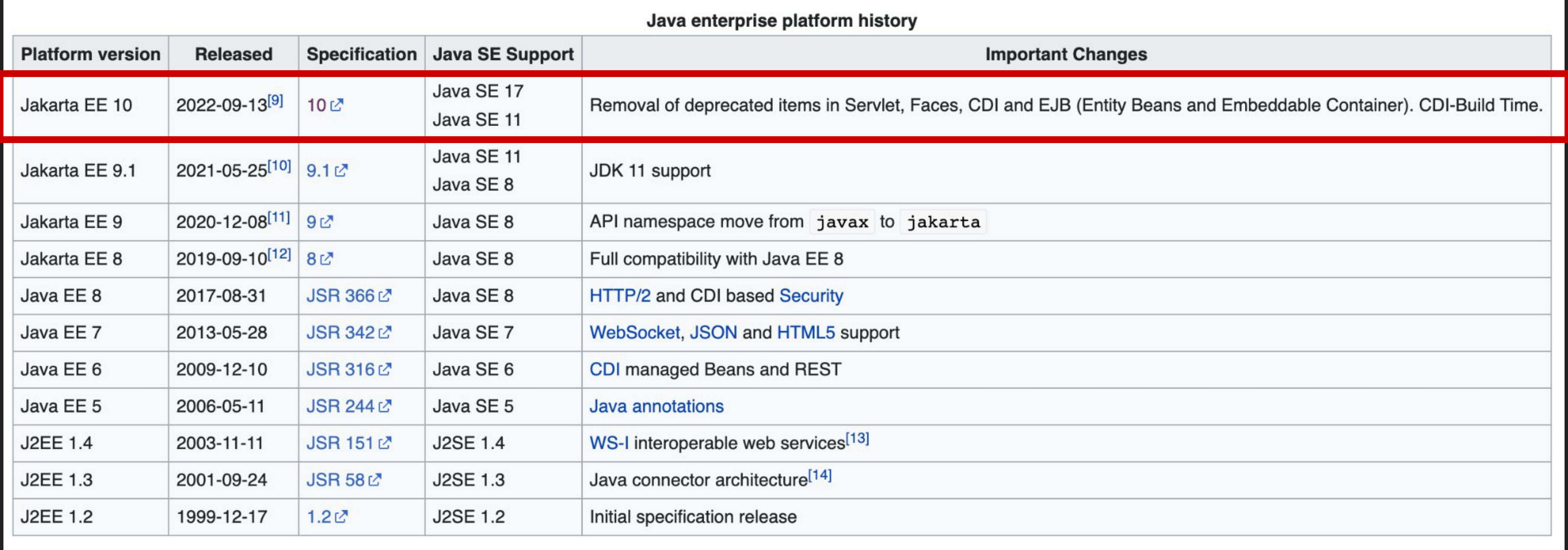


Java EE 용어		Jakarta EE 용어	
Java Servlet	javax.servlet	Jakarta Servlet	jakarta.servlet
JavaServer Pages (JSP)	javax.servlet.jsp	Jakarta Server Pages	jakarta.servlet.jsp
JavaServer Faces (JSF)	javax.faces	Jakarta Server Faces	jakarta.faces
Java Message Service (JMS)	javax.jms	Jakarta Messaging	jakarta.jms
Java Persistence API (JPA)	javax.persistence	Jakarta Persistence	jakarta.persistence
Java Transaction API (JTA)	javax.transaction	Jakarta Transaction	jakarta.transaction
Enterprise JavaBeans (EJB)	javax.ejb	Jakarta Enterprise Beans	jakarta.ejb
Java Mail	javax.mail	Jakarta Mail	Jakarta.mail

[표 1] 자바EE와 자카르타EE 용어 비교

https://www.samsungsds.com/kr/insights/java_jakarta.html

Jakarta EE - Jakarta EE 10



Java enterprise platform history

Platform version	Released	Specification	Java SE Support	Important Changes
Jakarta EE 10	2022-09-13[9]	10	Java SE 17 Java SE 11	Removal of deprecated items in Servlet, Faces, CDI and EJB (Entity Beans and Embeddable Container). CDI-Build Time.
Jakarta EE 9.1	2021-05-25[10]	9.1	Java SE 11 Java SE 8	JDK 11 support
Jakarta EE 9	2020-12-08[11]	9	Java SE 8	API namespace move from `javax` to `jakarta`
Jakarta EE 8	2019-09-10[12]	8	Java SE 8	Full compatibility with Java EE 8
Java EE 8	2017-08-31	JSR 366	Java SE 8	HTTP/2 and CDI based Security
Java EE 7	2013-05-28	JSR 342	Java SE 7	WebSocket, JSON and HTML5 support
Java EE 6	2009-12-10	JSR 316	Java SE 6	CDI managed Beans and REST
Java EE 5	2006-05-11	JSR 244	Java SE 5	Java annotations
J2EE 1.4	2003-11-11	JSR 151	J2SE 1.4	WS-I interoperable web services[13]
J2EE 1.3	2001-09-24	JSR 58	J2SE 1.3	Java connector architecture[14]
J2EE 1.2	1999-12-17	1.2	J2SE 1.2	Initial specification release

https://en.wikipedia.org/wiki/Jakarta_EE

스프링 프레임워크 6.0 살펴보기

Spring framework 6.0 Overview

스프링 프레임워크 6.0 - Overview

- Baseline
  - Java 17
  - Jakarta EE 10(Jakarta EE 9 지원: 네임스페이스 변경(javax -> jakarta)
- General Core
  - AOT(Ahead Of Time) 을 통한 실행속도 향상
  - PathMatchingResourcePatternResource: 사용 클래스패스 경로 자원 검색
  - DefaultFormattingConversionService: ISO 기반 java.time 유형 파싱
- Data Accessing and Transaction
  - JPA support for Hibernate ORM 6.1
  - Upgrade to R2DBC 1.0(R2DBC 트랜잭션 정의 포함)

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x/

스프링 프레임워크 6.0 - Overview

- **Spring Messaging**
    - RSocket interface client - @RSocketExchange 서비스 인터페이스
    - Netty 5 기반 Reactor Netty 2 지원
    - Jakarta WebSocket 2.1 및 표준 WebSocket 프로토콜 업그레이드 기작 지원
- **General Web**
    - HTTP interface client (@HttpExchange) 서비스 인터페이스
    - RFC 7807 - Problem Detail
    - HTTP status code 제어
    - Jackson 2.14 Support
    - Servlet 6.0 맞춤
- **Spring MVC**
    - PathPatternParser 기본 사용
    - Tiles 와 FreeMarker JSP 지원 제거

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x/

스프링 프레임워크 6.0 - Overview

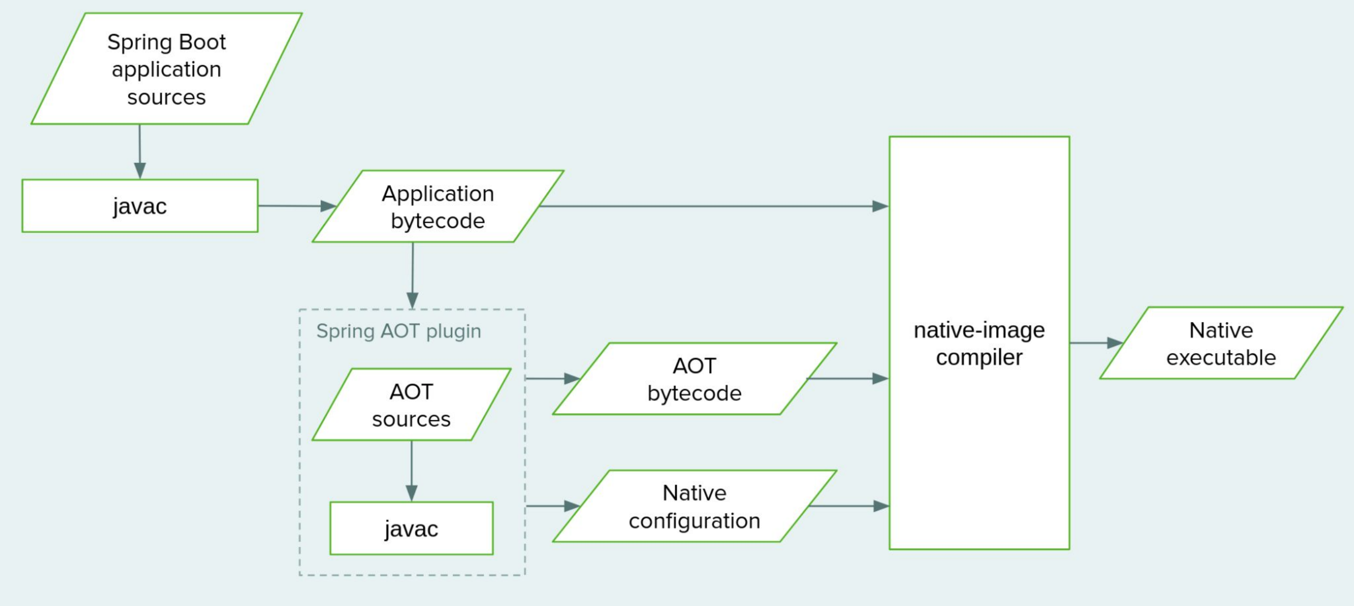
- Spring WebFlux
    - PartEvent - Stream multipart form upload
    - ResponseEntityExceptionHandler - WebFlux exception
    - Flux - non-streaming media types 반환
    - Netty 5 기반 Reactor Netty 2 지원
    - JDK HttpClient 를 통합한 WebClient
- Observability
    - Micrometer Observation 를 이용한 직접 관측 연결
- Testing
    - JVM 혹은 GraalVM 상에서 AOT 처리된 애플리케이션 컨텍스트 테스트 지원
    - HtmlUnit 2.64 요청 파라미터 제어 통합
    - Servlet API 6.0 을 기반한 서블릿 Mock(MockHttpServletRequest, MockHttpSession) 지원

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x/

스프링 프레임워크 6.0 - General Core

GraalVM Native Image Support

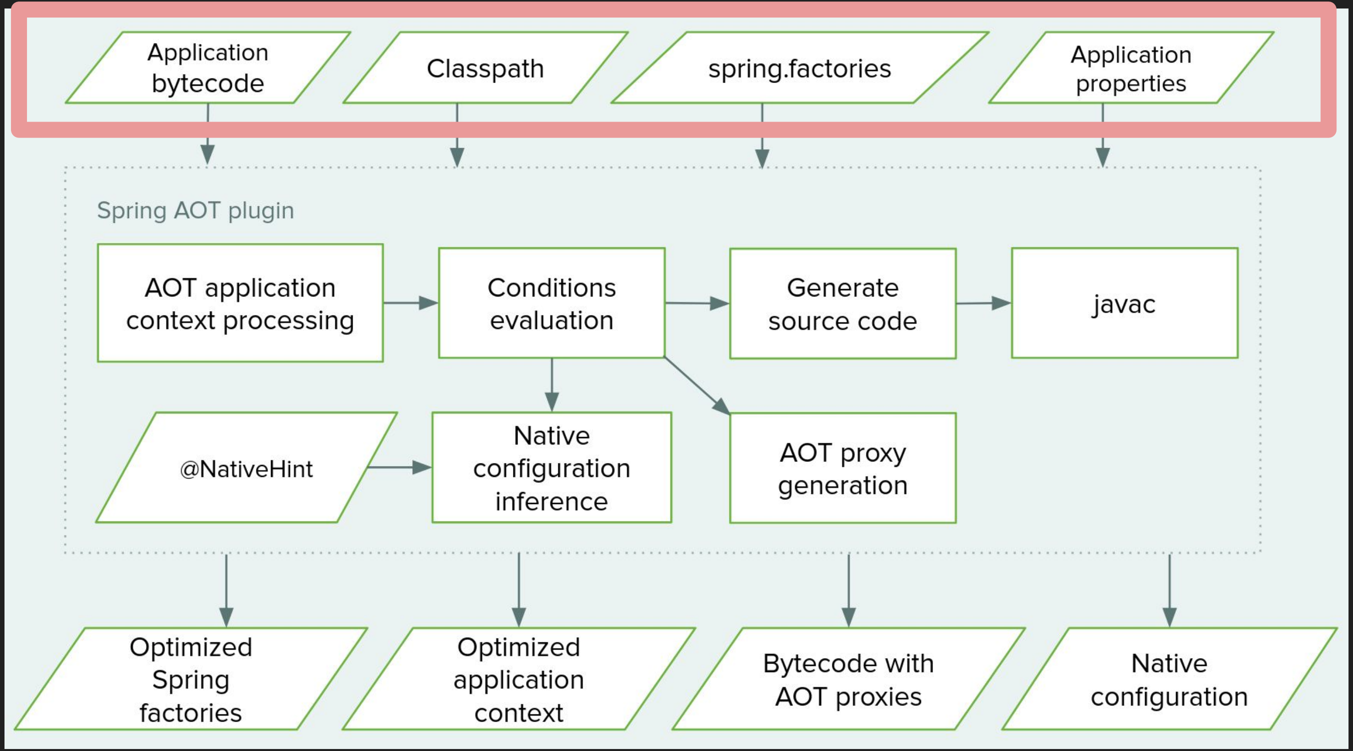
스프링 애플리케이션을 심층분석하여 GraalVM 에 맞춰 애플리케이션을 변환하고 최적화



https://docs.spring.io/spring-boot/docs/current/reference/html/native-image.html

스프링 프레임워크 6.0 - General Core

Native Configuration 최적화(그레이들, 메이븐 플러그인 제공)



https://docs.spring.io/spring-boot/docs/current/reference/html/native-image.html

스프링 프레임워크 6.0 - General Core

Native Image 도입...?

- JVM 설치없이 실행이 가능하고
- 실행시 인프라 소모가 줄어들고
- 구동시간을 줄일 수 있고
- 빈 등록과정을 프로그래밍하여 리플렉션을 줄일 수 있다.

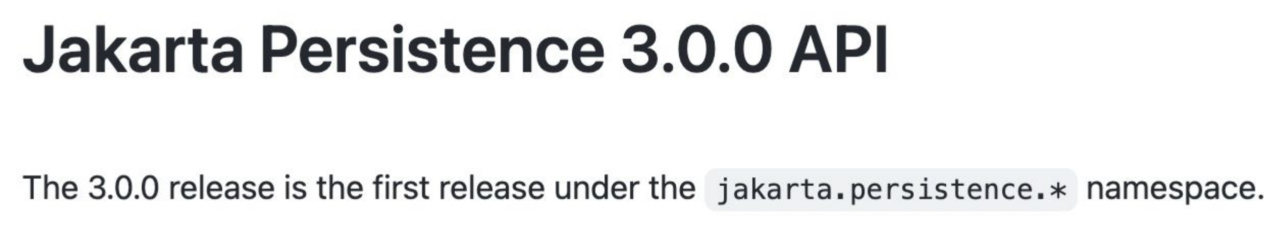
... 뭔가 해야할 게 많은 것 같다.

... 당장 써야겠다는 마음은 없다.

https://docs.spring.io/spring-boot/docs/current/reference/html/native-image.html

스프링 프레임워크 6.0 - Data Access And Transaction

Jakarta Persistence 3.0.0



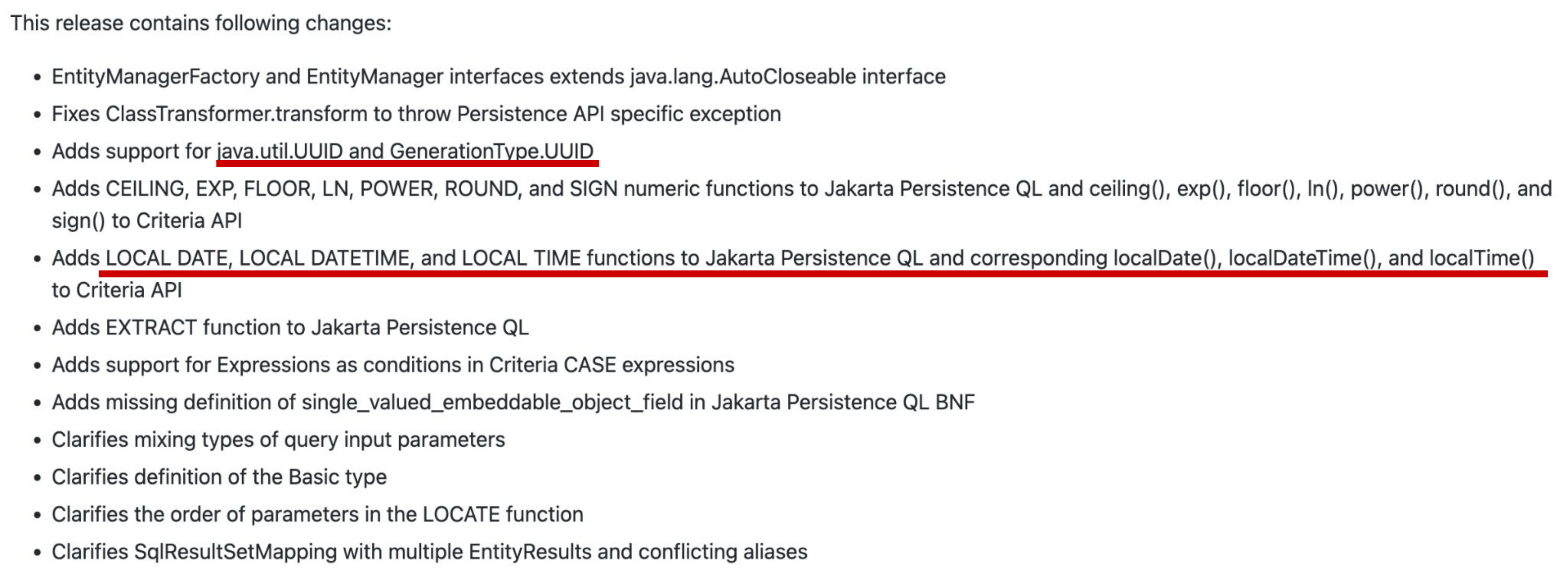
Jakarta Persistence 3.0.0 API

The 3.0.0 release is the first release under the `jakarta.persistence.*` namespace.

https://github.com/jakartaee/persistence/releases/tag/3.0-3.0.0-RELEASE

스프링 프레임워크 6.0 - Data Access And Transaction

Jakarta Persistence 3.1.0



This release contains following changes:

- EntityManagerFactory and EntityManager interfaces extends java.lang.AutoCloseable interface
- Fixes ClassTransformer.transform to throw Persistence API specific exception
- Adds support for java.util.UUID and GenerationType.UUID
- Adds CEILING, EXP, FLOOR, LN, POWER, ROUND, and SIGN numeric functions to Jakarta Persistence QL and ceiling(), exp(), floor(), ln(), power(), round(), and sign() to Criteria API
- Adds LOCAL DATE, LOCAL DATETIME, and LOCAL TIME functions to Jakarta Persistence QL and corresponding localDate(), localDateTime(), and localTime() to Criteria API
- Adds EXTRACT function to Jakarta Persistence QL
- Adds support for Expressions as conditions in Criteria CASE expressions
- Adds missing definition of single_valued_embeddable_object_field in Jakarta Persistence QL BNF
- Clarifies mixing types of query input parameters
- Clarifies definition of the Basic type
- Clarifies the order of parameters in the LOCATE function
- Clarifies SqlResultSetMapping with multiple EntityResults and conflicting aliases

https://github.com/jakartaee/persistence/releases/tag/3.1-3.1.0-RELEASE

스프링 프레임워크 6.0 - Data Access And Transaction

UUID를 대체하는! TSID(Time-Sorted Unique Identifiers)

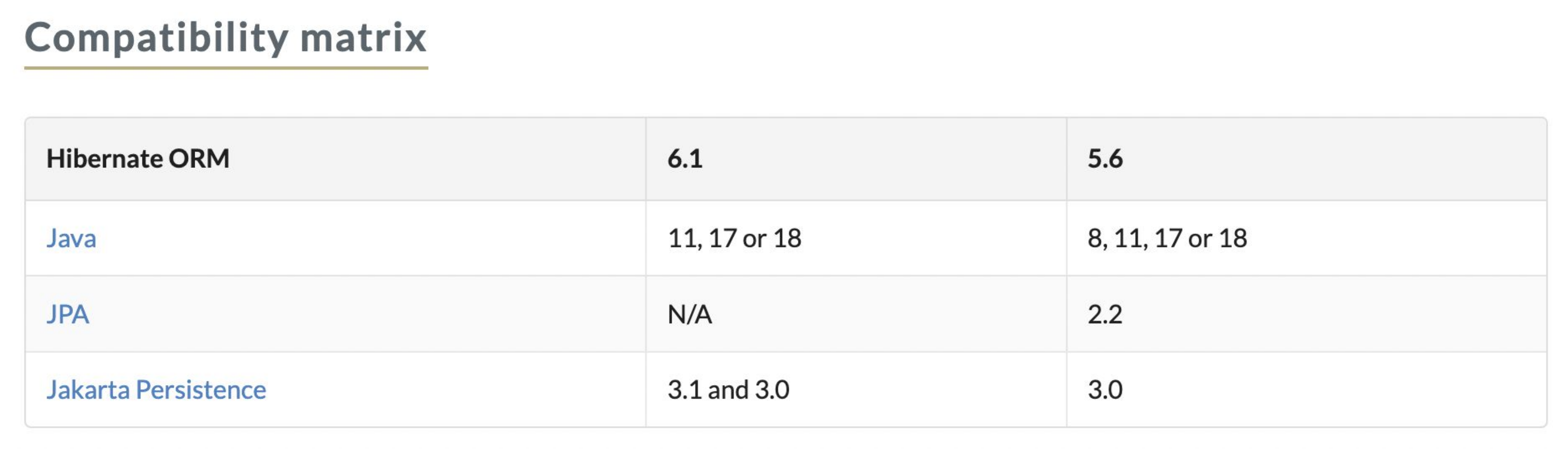
- TSID numerical value(18): 388400145978465528
- TSID string value(13): 0ARYZVZXW377R
- TSID time millis since epoch value(13): 1670438610927

https://vladmihalcea.com/uuid-database-primary-key/

스프링 프레임워크 6.0 - Data Access And Transaction

Hibernate ORM 6.1

- JPA 2.x -> 3.0 (JPA -> Jakarta Persistence)
- Jakarta EE 9 지원: javax.persistence -> jakarta.persistence



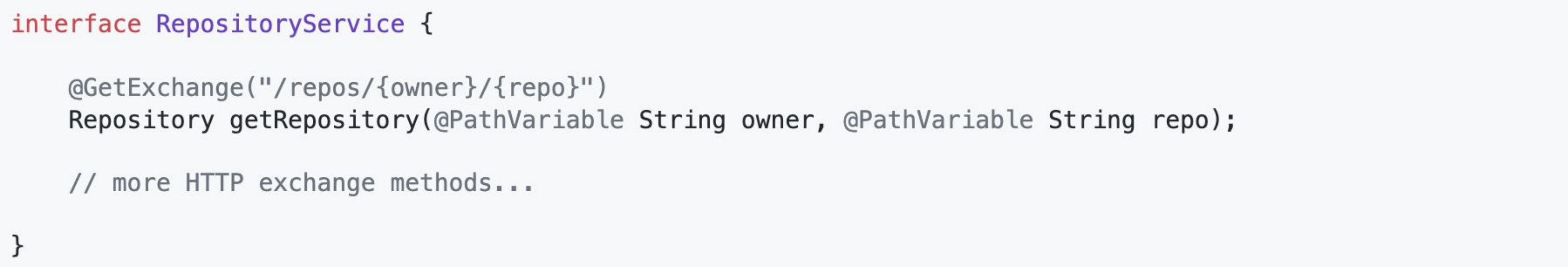
Compatibility matrix

Hibernate ORM	6.1	5.6
Java	11, 17 or 18	8, 11, 17 or 18
JPA	N/A	2.2
Jakarta Persistence	3.1 and 3.0	3.0

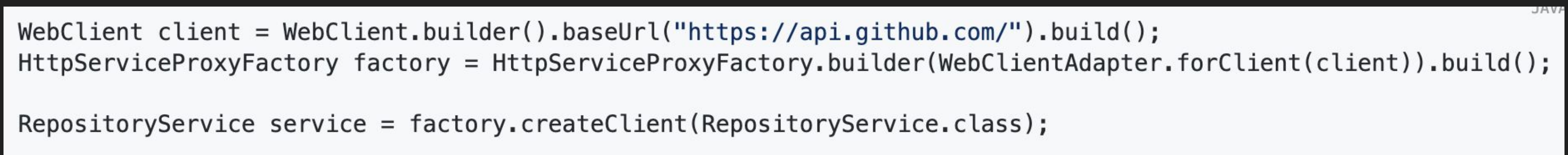
https://vladmihalcea.com/uuid-database-primary-key/

스프링 프레임워크 6.0 - General Web

HTTP interface(feign과 유사) - 프록시 생성해서 사용가능...!! —0-)!



```java
interface RepositoryService {

    @GetExchange("/repos/{owner}/{repo}")
    Repository getRepository(@PathVariable String owner, @PathVariable String repo);

    // more HTTP exchange methods...

}
```



```java
WebClient client = WebClient.builder().baseUrl("https://api.github.com/").build();
HttpServiceProxyFactory factory = HttpServiceProxyFactory.builder(WebClientAdapter.forClient(client)).build();

RepositoryService service = factory.createClient(RepositoryService.class);
```

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#general-web-revision

스프링 프레임워크 6.0 - General Web

RestTemplate && WebClient && HTTP interface

RestTemplate은 스프링에서 제거되나요?

토비의 스프링 이야기

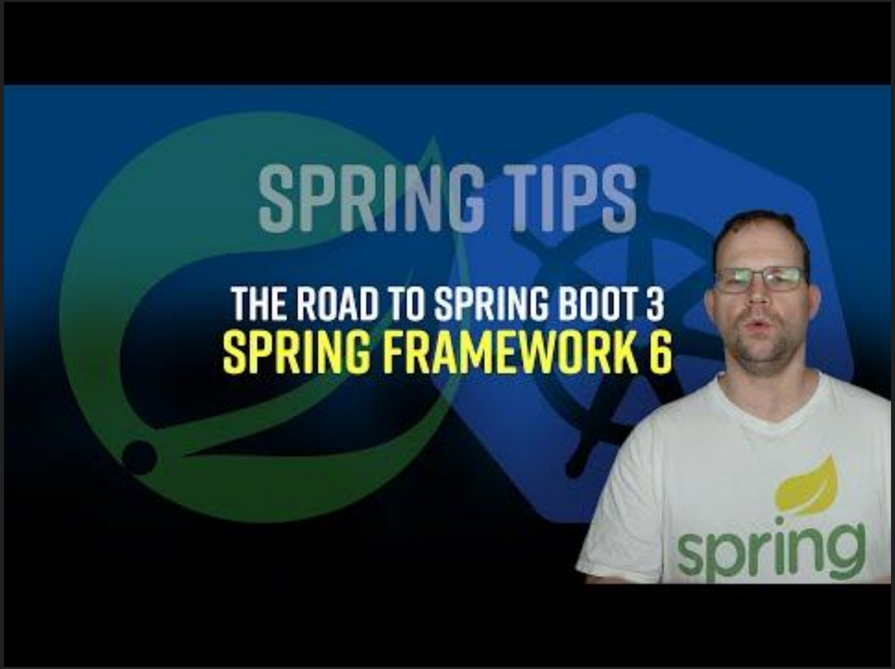
Spring 6의 새로운 HTTP Interface와 3가지 Rest Clients 라이브 코딩

토비의 스프링 이야기

https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#webmvc-client

스프링 프레임워크 6.0 - General Web

RFC 7807 - 'Problem Details for HTTP APIs'



https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#general-web-revision

스프링 프레임워크 6.0 - General Web

Jackson 2.14.1

- https://github.com/FasterXML/jackson/wiki/Jackson-Release-2.14

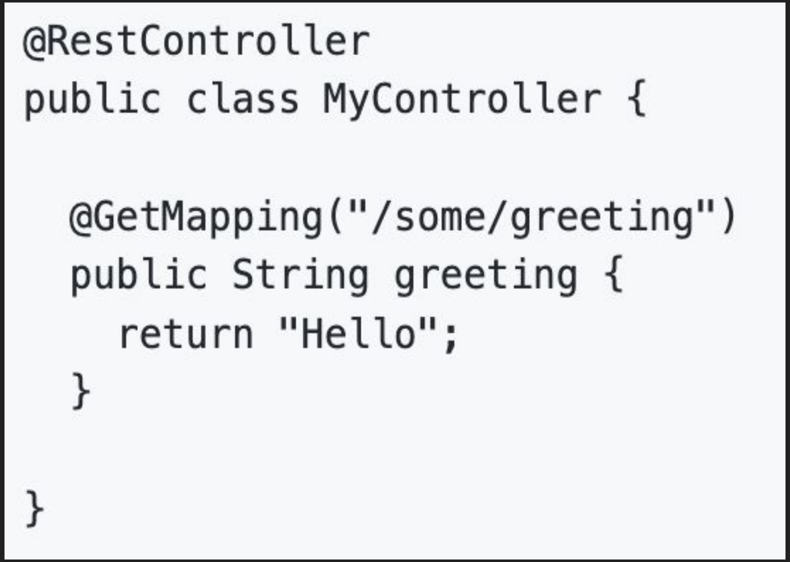
Jakarta Servlet 6.0

- https://jakarta.ee/specifications/servlet/6.0/

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#general-web-revision

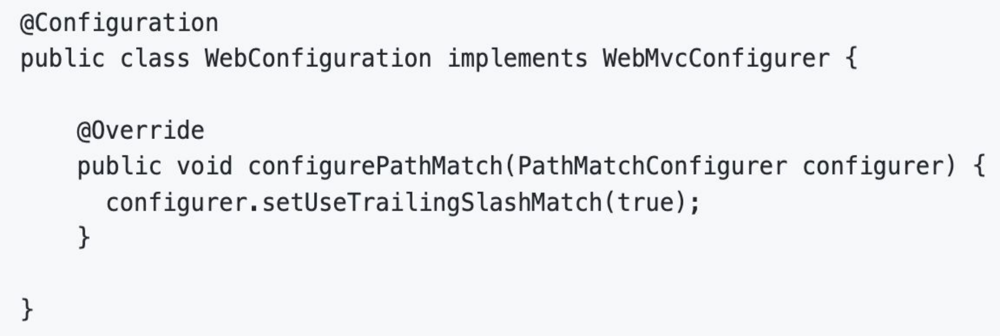
스프링 프레임워크 6.0 - Spring WebMVC

PathPatternParser - 후행 슬래시(trailing slash, URL 마지막 /) 허용하지 않음



```
@RestController
public class MyController {

  @GetMapping("/some/greeting")
  public String greeting {
    return "Hello";
  }

}
```

GET /some/greeting/ != GET /some/greeting



```
@Configuration
public class WebConfiguration implements WebMvcConfigurer {

    @Override
    public void configurePathMatch(PathMatchConfigurer configurer) {
      configurer.setUseTrailingSlashMatch(true);
    }

}
```

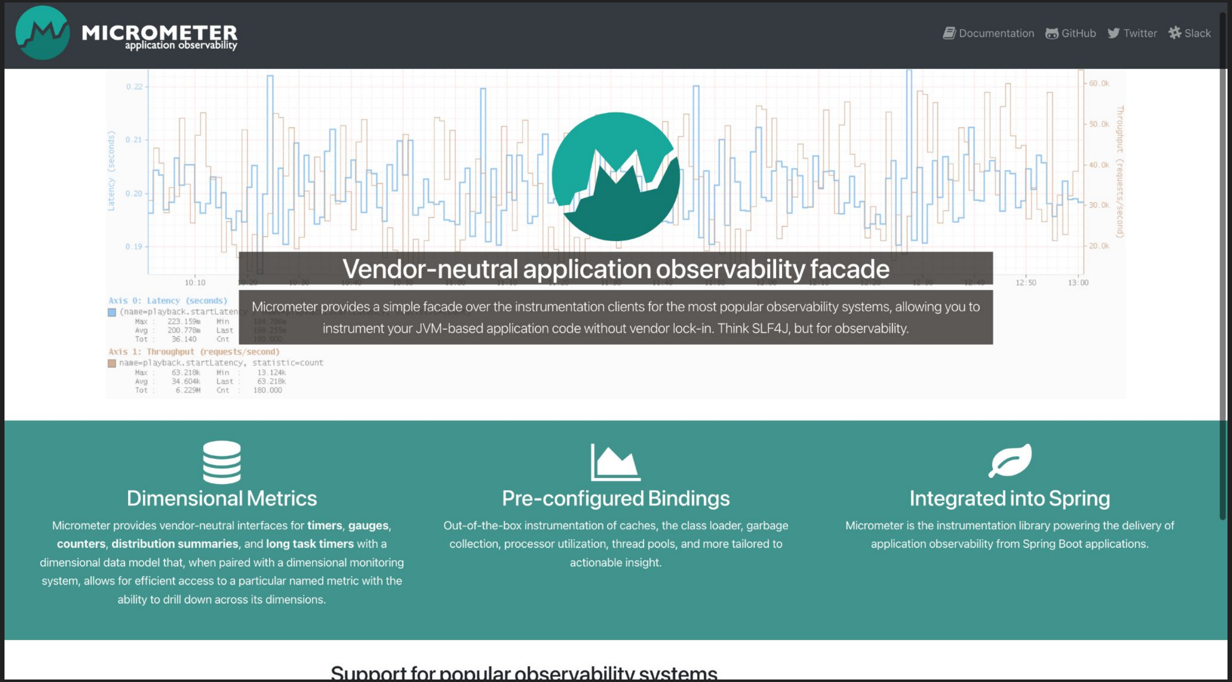
임시로 이렇게 사용

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#spring-mvc

스프링 프레임워크 6.0 - Observability

Spring Framework 여러 부분에서 Micrometer Observation 사용한 직접계측

- spring-web 은 io.micrometer:micrometer-observation 의존성 가짐


MICROMETER
application observability
Documentation
GitHub
Twitter
Slack
0.22
0.21
0.20
0.19
Latency (seconds)
60.0k
50.0k
40.0k
30.0k
20.0k
Throughput (requests/second)
10:10
12:50
13:00
Axis 0: Latency (seconds)
(name=playback.startLatency
Max : 223.159m Min
Avg : 200.778m Last
Tot : 36.140 Cnt
Axis 1: Throughput (requests/second)
name=playback.startLatency, statistic=count
Max : 63.218k Min : 13.124k
Avg : 34.604k Last : 63.218k
Tot : 6.229M Cnt : 180.000

Vendor-neutral application observability facade

Micrometer provides a simple facade over the instrumentation clients for the most popular observability systems, allowing you to instrument your JVM-based application code without vendor lock-in. Think SLF4J, but for observability.

Dimensional Metrics

Micrometer provides vendor-neutral interfaces for **timers**, **gauges**, **counters**, **distribution summaries**, and **long task timers** with a dimensional data model that, when paired with a dimensional monitoring system, allows for efficient access to a particular named metric with the ability to drill down across its dimensions.

Pre-configured Bindings

Out-of-the-box instrumentation of caches, the class loader, garbage collection, processor utilization, thread pools, and more tailored to actionable insight.

Integrated into Spring

Micrometer is the instrumentation library powering the delivery of application observability from Spring Boot applications.

Support for popular observability systems

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#observability

스프링 프레임워크 6.0 - Testing

- JVM 혹은 GraalVM 상에서 AOT 처리된 애플리케이션 컨텍스트 테스트 지원
- HtmlUnit 2.64 요청 파라미터 제어 통합
- Servlet API 6.0 을 기반한 서블릿 Mock(MockHttpServletRequest, MockHttpSession) 지원

https://github.com/spring-projects/spring-framework/wiki/What%27s-New-in-Spring-Framework-6.x#testing

스프링 프레임워크 6.0 - Upgrade

Jakarta EE 9 적용되면서 변경되는 네임스페이스 변경 신경쓰기

- javax.* -> jakarta.*
- javax.inject -> jakarta.inject
- javax.annotation -> jakarta.annotation
- javax.persistence - > jakarta.persistence
- javax.servlet -> jakarta.servlet

Java 버전 먼저 17로 올리고!
그다음 스프링 프레임워크 업그레이드!

https://github.com/spring-projects/spring-framework/wiki/Upgrading-to-Spring-Framework-6.x

스프링 부트 3.0 살펴보기

Spring Boot 3.0 Overview

스프링 부트 3.0 - 최소 요구사양

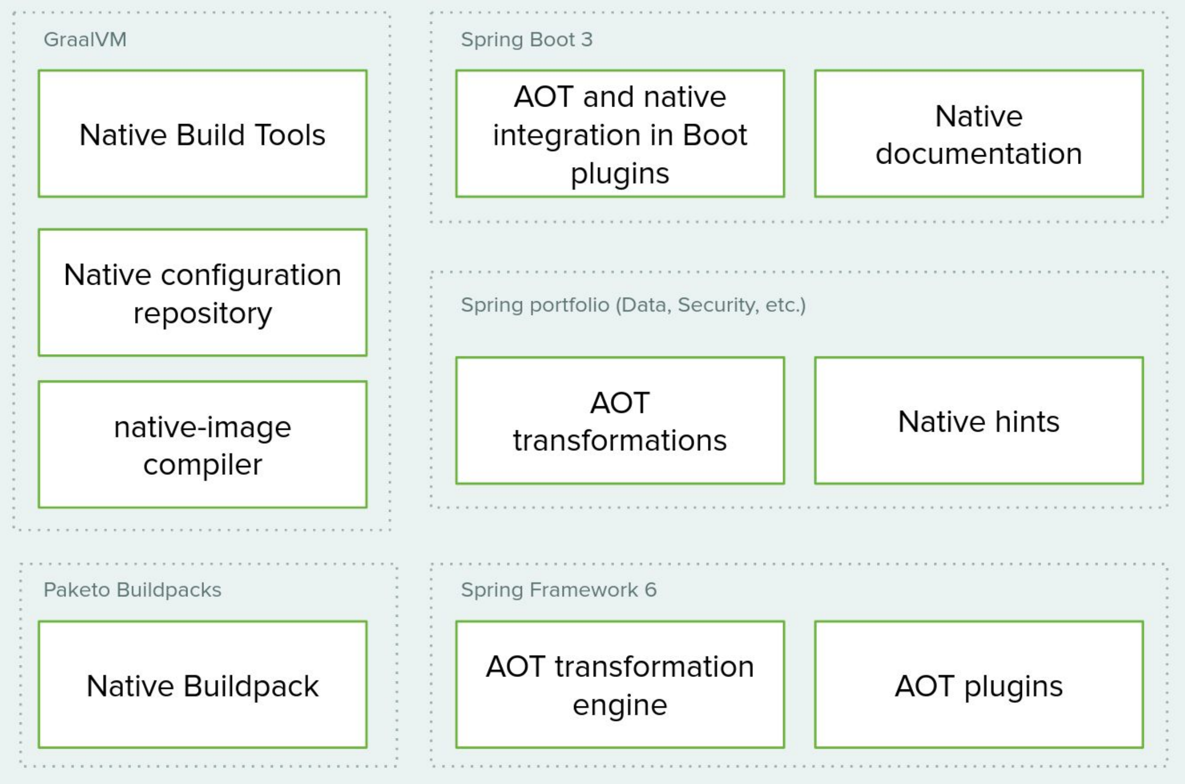
- 최소 요구사양 변경
  - Gradle 7.3
  - Java 17
  - Kotlin 1.6
  - Jakarta EE 9
  - Spring Framework 6

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0-Release-Notes

스프링 부트 3.0

// First-class Native Support

스프링 부트 3.0 - First-class Native Support



https://spring.io/blog/2021/12/09/new-aot-engine-brings-spring-native-to-the-next-level

Spring Boot 3.0.0-M1(2022-01-21)

- Jakarta EE 9 적용
- Jakarta EE 9 미지원 라이브러리 중단 및 제거
  - Apache ActiveMQ
  - Atomikos
  - EhCache 2
  - Hazelcast 3
- CommonsMultipartResolver 지원 중단 및 제거
  - StandardMultipartResolver 기본

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M1-Release-Notes

Spring Boot 3.0.0-M2(2022-03-25)

- Groovy 4.0
- RxJava 1.x 와 RxJava 2.x 의존성관리 제거 및 RxJava 3 대체
- 이미지 배너 지원 제거(텍스트 배너만 지원)
- Micrometer Observability 지원
- Jakarta EE 9 지원 복귀
    - EhCache 3
    - H2Console
- @ConstructorBinding 기능 향상: 생성자가 2개이상인 경우 선언필요

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M2-Release-Notes

Spring Boot 3.0.0-M3(2022-05-20)

- Actuator 계측노출 속성 변경
  - **old:** management.metrics.export.<product> - deprecated
  - **new:** management.<product>.metrics.export
  - spring-boot-properties-migrator **사용시 알림**
- **메인클래스명 탐색 간결화 - Gradle**
  - bootJar, bootRun, bootWar **에서 메인 클래스 지정시 발생할 오류 대체**
  - **springBoot DSL 추가하여 메인클래스 정의**
- **Spring Security - SAML2 Relying Part 구성 변경**
- **내장형 MongoDB 지원 제거됨**
- **로깅 일자 형식 변경(ISO-8601)** yyyy-MM-dd'T'HH:mm:ss.SSSXXX

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M3-Release-Notes

Spring Boot 3.0.0-M3(2022-05-20)

- Apache Solr 지원 제거
- Micrometer's JvmInfoMetrics 자동구성
- Jakarta EE 9 지원에 따라 복귀
    - REST Assured(5.0)
    - Pooled JMS(Apache Artemis)
- Micrometer Observability
    - Micrometer Observation API 자동구성
    - Micrometer Tracing 자동구성
    - Micrometer OtlpMeterRegistory
- Apache Kafaka 비동기 Acks 활성화
    - spring.kafka.listener.async-acks: true
    - spring.kafka.listener.async-mode: manual | immediate

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M3-Release-Notes

Spring Boot 3.0.0-M4(2022-07-21)

- **후행 슬래시(trailing slash) 매칭 구성 비활성화**
  - GET /some/greeting != GET /some/greeting/
- **Hibernate ORM 6.1**
  - **spring-boot-starter-data-jpa 내에 Hibernate Group ID Change**
  - org.hibernate:hibernate-core -> org.hibernate.orm:hibernate-core
- Flyway 9.0 지원
- R2DBC 1.0 지원
- Elasticsearch Clients Templates
- Reactor Netty가 없으면 JdkClientHttpConnector가 자동구성되어 WebClient이 사용함

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M4-Release-Notes

Spring Boot 3.0.0-M5(2022-09-23)

- spring.data 접두사 - Spring Data 전용 확정
    - Cassandra: spring.data.cassandra.* -> spring.cassandra.*
    - Redis: spring.redis.* -> spring.data.redis.*
- JMX 종단점(Endpoint) 노출: health 만 노출
- 액츄에이터 종단점 민감데이터 기본 마스킹 처리 및 속성 설정
- Flyway 구성 사용자정의 지원
- @EnableBatchProcessing 필요없음 자동인식됨
- MongoDB Health Check 제공
- spring.factories 를 이용한 자동구성 등록지원 기능 제거
- Spring Native 지원
- @SpringBootTest 에서 메인메서드 사용여부 결정 가능

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-M5-Release-Notes

Spring Boot 3.0.0-RC1(2022-10-21)

- server.max-http-header-size: 웹컨테이너 일괄적용
- Graceful Shutdown 에 사용되는 SmartLifeCycle 페이즈 변경
- 스프링 부트 태스크 Properties 이용해서 정의 가능
- build-info.properties 항목 중 속성 제외가능
- Spring Native -> GraalVM Native
- Spring Data JDBC 자동구성 지원

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-RC1-Release-Notes

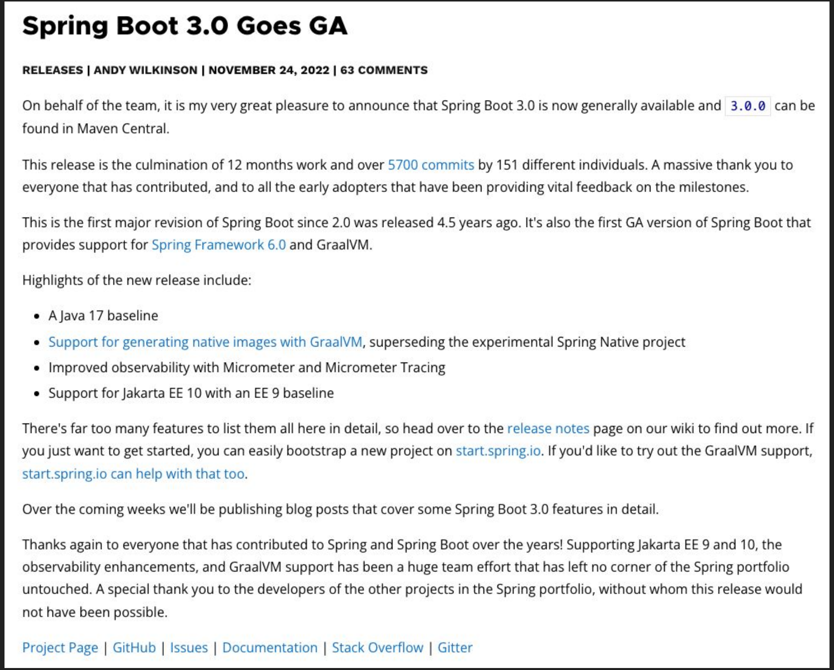
Spring Boot 3.0.0-RC2(2022-11-11)

- Jakarta EE 10 지원
- Jetty 사용 시 Servlet 지원여부 확인하여 조정
- httptrace 종단명 변경: httpexchange
- 액츄에이터 종단점 별로 ObjectMapper 사용 가능
- Spring Native 사용시 Graal 22.3 혹은 최신

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0.0-RC2-Release-Notes

Spring Boot 3.0.0 GA(2022-11-24)

- 스프링 부트 2.0 출시 후 4.5년!
- Spring Framework 6 과 GraalVM 지원(정식버전은...)



Spring Boot 3.0 Goes GA

RELEASES | ANDY WILKINSON | NOVEMBER 24, 2022 | 63 COMMENTS

On behalf of the team, it is my very great pleasure to announce that Spring Boot 3.0 is now generally available and `3.0.0` can be found in Maven Central.

This release is the culmination of 12 months work and over 5700 commits by 151 different individuals. A massive thank you to everyone that has contributed, and to all the early adopters that have been providing vital feedback on the milestones.

This is the first major revision of Spring Boot since 2.0 was released 4.5 years ago. It's also the first GA version of Spring Boot that provides support for Spring Framework 6.0 and GraalVM.

Highlights of the new release include:

- A Java 17 baseline
- Support for generating native images with GraalVM, superseding the experimental Spring Native project
- Improved observability with Micrometer and Micrometer Tracing
- Support for Jakarta EE 10 with an EE 9 baseline

There's far too many features to list them all here in detail, so head over to the release notes page on our wiki to find out more. If you just want to get started, you can easily bootstrap a new project on start.spring.io. If you'd like to try out the GraalVM support, start.spring.io can help with that too.

Over the coming weeks we'll be publishing blog posts that cover some Spring Boot 3.0 features in detail.

Thanks again to everyone that has contributed to Spring and Spring Boot over the years! Supporting Jakarta EE 9 and 10, the observability enhancements, and GraalVM support has been a huge team effort that has left no corner of the Spring portfolio untouched. A special thank you to the developers of the other projects in the Spring portfolio, without whom this release would not have been possible.

Project Page | GitHub | Issues | Documentation | Stack Overflow | Gitter

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0-Release-Notes

정리

Java

- Java 11을 넘어서 17까지!
- 점점 제공되는 기능이 풍부해지고 있음
- 코틀린...? 굳이 자바에서 코틀린으로 옮겨갈 필요성이 줄어들고 있음

정리

Spring Framework 6.0

- Java EE 8 -> Jakarta EE 9 변경
- 네임스페이스(namespace) 변경: javax.* -> jakarta.*
- AOT(Ahead-Of-Time) 엔진 도입
  - Spring Native 이미지(Graal VM image) 빌드
  - 빌드 단계에서 정적분석을 통해서 최적화 작업 수행
- Micrometer 관측 지원 강화
- 실험실에서 연구되던 기능들 반영
  - Spring Native(GraalVM)

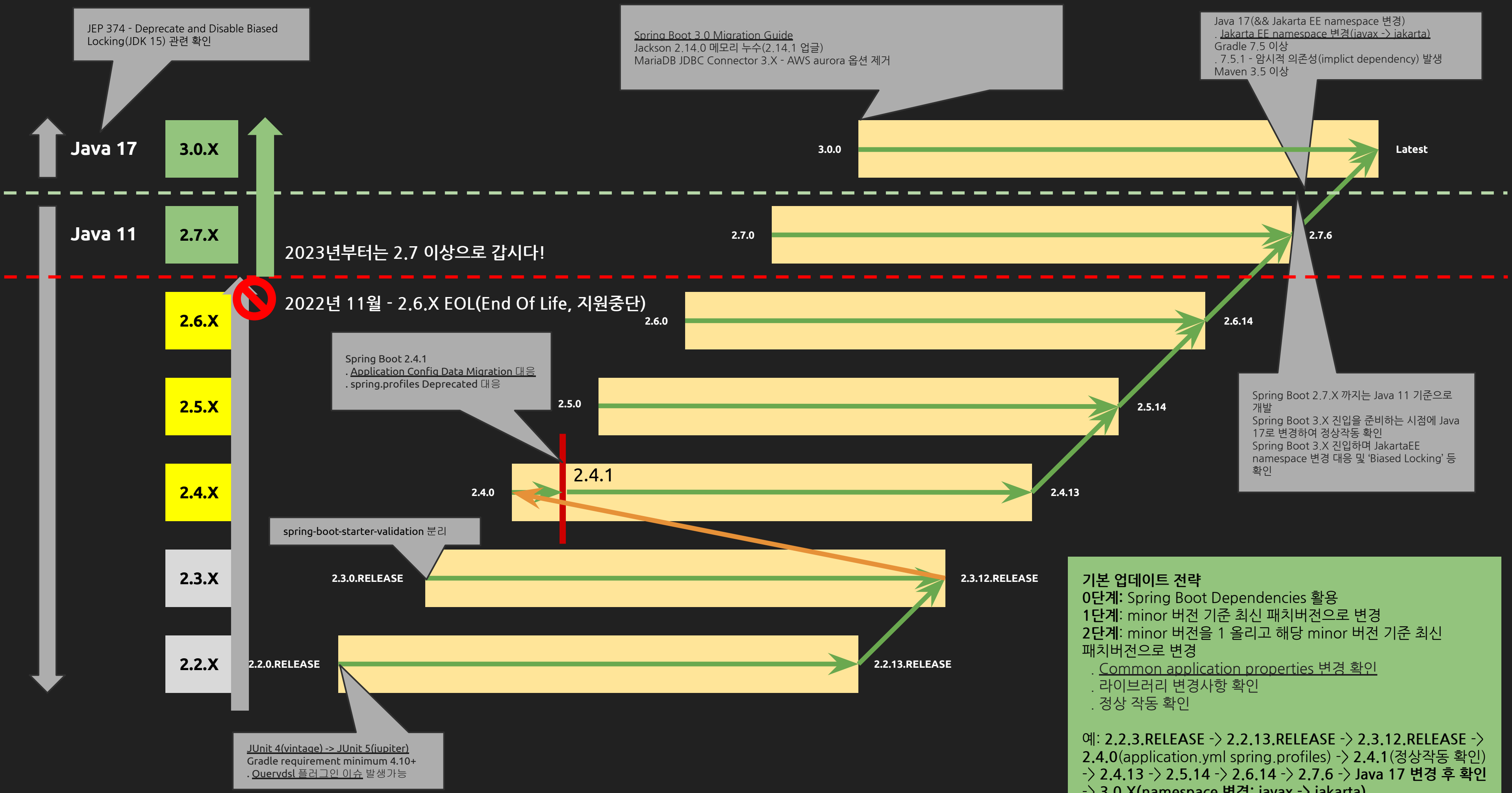
정리

Spring Boot 3.0

- Java 17과 Kotlin 1.6 이상 지원
- Spring Framework 6.0 적용
  - Jakarta EE 9 최적화
- 그레이들 7.X(7.5 이상)
- 미사용 라이브러리 및 지원 중단

스프링 부트 업그레이드 전략

Spring Boot Upgrade Strategy



JEP 374 - Deprecate and Disable Biased
Locking(JDK 15) 관련 확인

Spring Boot 3.0 Migration Guide
Jackson 2.14.0 메모리 누수(2.14.1 업글)
MariaDB JDBC Connector 3.X - AWS aurora 옵션 제거

Java 17(&& Jakarta EE namespace 변경)
. Jakarta EE namespace 변경(javax -> jakarta)
Gradle 7.5 이상
. 7.5.1 - 암시적 의존성(implict dependency) 발생
Maven 3.5 이상

Java 17 **3.0.X** 3.0.0 Latest

Java 11 **2.7.X** 2.7.0 2.7.6

2023년부터는 2.7 이상으로 갑시다!

2.6.X 2022년 11월 - 2.6.X EOL(End Of Life, 지원중단) 2.6.0 2.6.14

2.5.X 2.5.0 2.5.14

Spring Boot 2.4.1
. Application Config Data Migration 대응
. spring.profiles Deprecated 대응

2.4.X 2.4.0 2.4.1 2.4.13

spring-boot-starter-validation 분리

2.3.X 2.3.0.RELEASE 2.3.12.RELEASE

2.2.X 2.2.0.RELEASE 2.2.13.RELEASE

JUnit 4(vintage) -> JUnit 5(jupiter)
Gradle requirement minimum 4.10+
. Querydsl 플러그인 이슈 발생가능

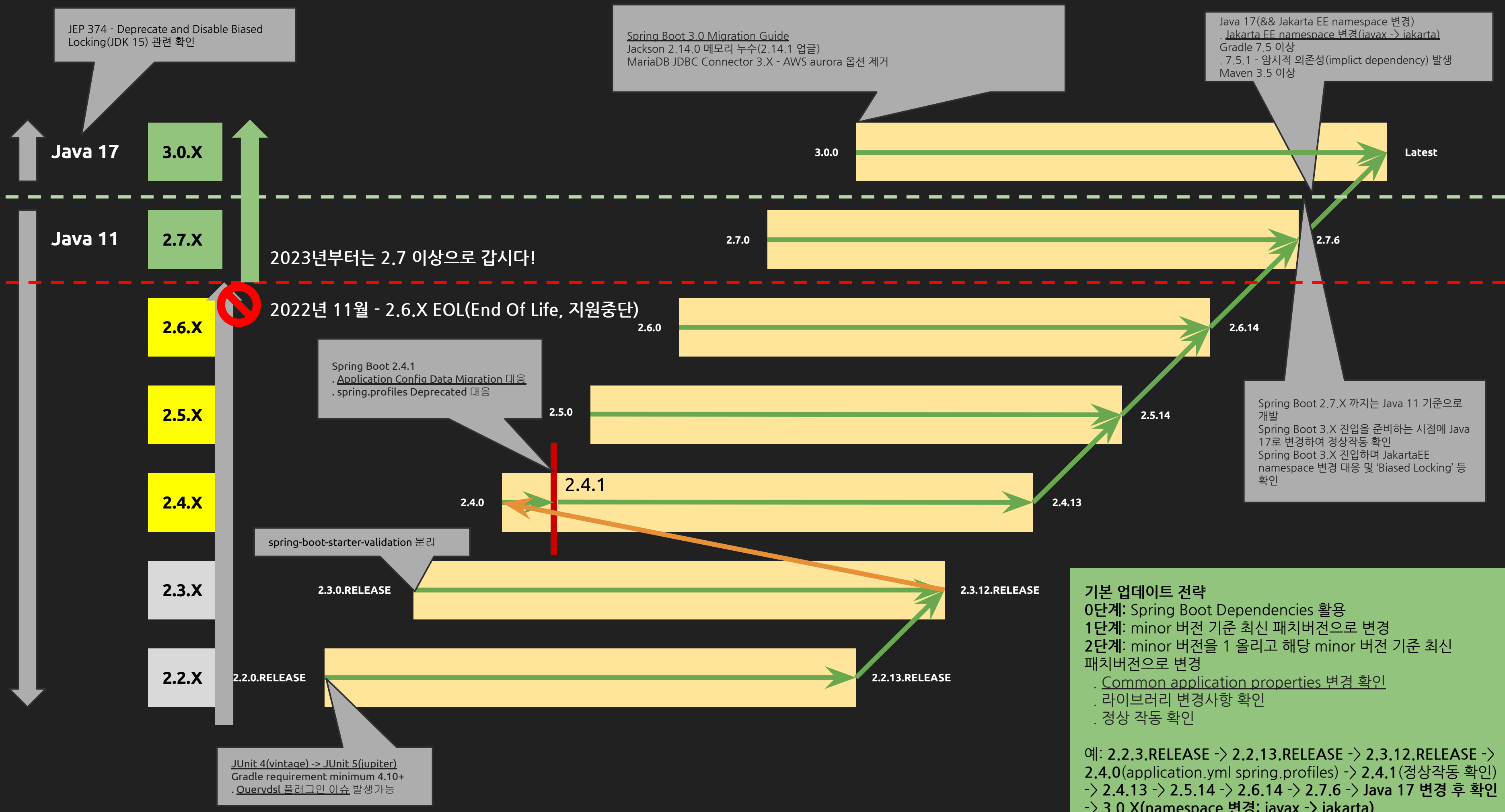
Spring Boot 2.7.X 까지는 Java 11 기준으로 개발
Spring Boot 3.X 진입을 준비하는 시점에 Java 17로 변경하여 정상작동 확인
Spring Boot 3.X 진입하며 JakartaEE namespace 변경 대응 및 'Biased Locking' 등 확인

기본 업데이트 전략
0단계: Spring Boot Dependencies 활용
1단계: minor 버전 기준 최신 패치버전으로 변경
2단계: minor 버전을 1 올리고 해당 minor 버전 기준 최신 패치버전으로 변경
. Common application properties 변경 확인
. 라이브러리 변경사항 확인
. 정상 작동 확인

예: **2.2.3.RELEASE** -> **2.2.13.RELEASE** -> **2.3.12.RELEASE** -> **2.4.0**(application.yml spring.profiles) -> **2.4.1**(정상작동 확인) -> **2.4.13** -> **2.5.14** -> **2.6.14** -> **2.7.6** -> **Java 17 변경 후 확인** -> **3.0.X(namespace 변경: javax -> jakarta)**

끗!

이지만 not 끗...



JEP 374 - Deprecate and Disable Biased Locking(JDK 15) 관련 확인

Spring Boot 3.0 Migration Guide
Jackson 2.14.0 메모리 누수(2.14.1 업글)
MariaDB JDBC Connector 3.X - AWS aurora 옵션 제거

Java 17(&& Jakarta EE namespace 변경)
. Jakarta EE namespace 변경(javax -> jakarta)
Gradle 7.5 이상
. 7.5.1 - 암시적 의존성(implict dependency) 발생
Maven 3.5 이상

2023년부터는 2.7 이상으로 갑시다!

2022년 11월 - 2.6.X EOL(End Of Life, 지원중단)

Spring Boot 2.4.1
. Application Config Data Migration 대응
. spring.profiles Deprecated 대응

Spring Boot 2.7.X 까지는 Java 11 기준으로 개발
Spring Boot 3.X 진입을 준비하는 시점에 Java 17로 변경하여 정상작동 확인
Spring Boot 3.X 진입하며 JakartaEE namespace 변경 대응 및 'Biased Locking' 등 확인

spring-boot-starter-validation 분리

JUnit 4(vintage) -> JUnit 5(jupiter)
Gradle requirement minimum 4.10+
. Querydsl 플러그인 이슈 발생가능

기본 업데이트 전략
0단계: Spring Boot Dependencies 활용
1단계: minor 버전 기준 최신 패치버전으로 변경
2단계: minor 버전을 1 올리고 해당 minor 버전 기준 최신 패치버전으로 변경
. Common application properties 변경 확인
. 라이브러리 변경사항 확인
. 정상 작동 확인

예: **2.2.3.RELEASE** -> **2.2.13.RELEASE** -> **2.3.12.RELEASE** -> **2.4.0**(application.yml spring.profiles) -> **2.4.1**(정상작동 확인) -> **2.4.13** -> **2.5.14** -> **2.6.14** -> **2.7.6** -> **Java 17 변경 후 확인** -> **3.0.X(namespace 변경: javax -> jakarta)**

짚고 넘어가기! 유의적 버전(Semantic Version)

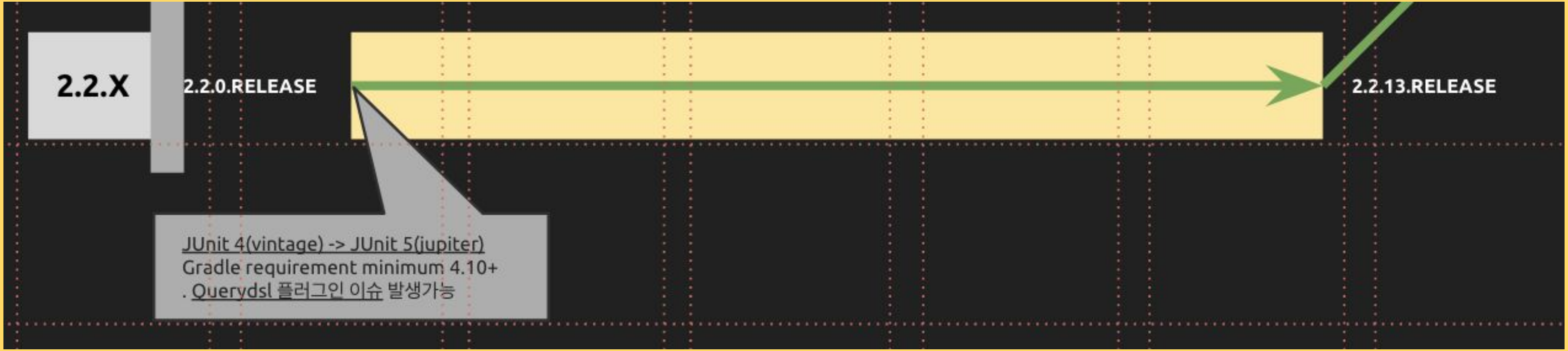
{Major}.{Minor}.{Patch}

주.부.수

엄청 귀찮은 버전업!

일단 하고 보는 버전업!

뭔가 귀찮을 수 있는 버전업!



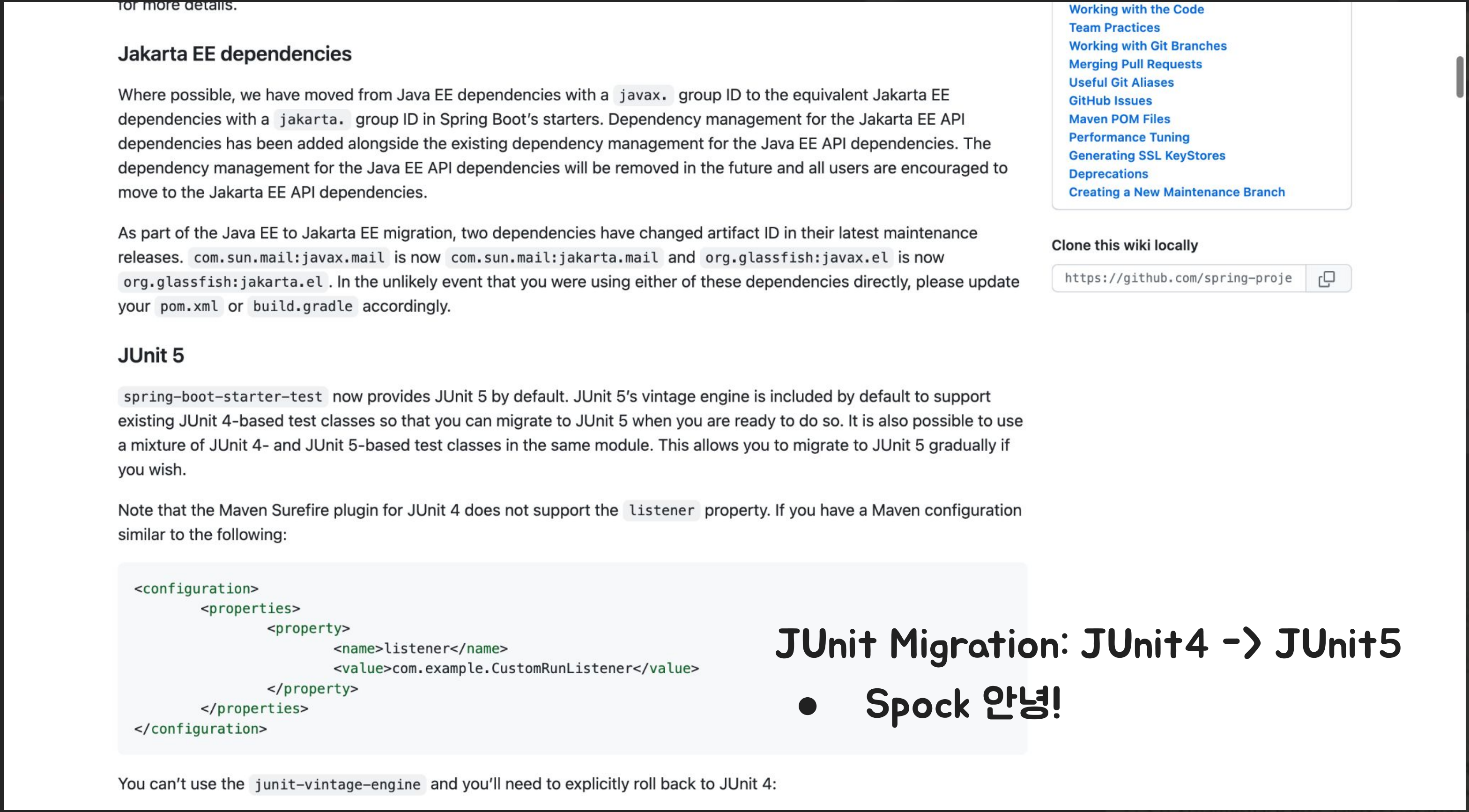
2.2.X

2.2.0.RELEASE

2.2.13.RELEASE

JUnit 4(vintage) -> JUnit 5(jupiter)
Gradle requirement minimum 4.10+
. Querydsl 플러그인 이슈 발생가능

spring-boot-starter:2.2.X.RELEASE



for more details.

Jakarta EE dependencies

Where possible, we have moved from Java EE dependencies with a `javax.` group ID to the equivalent Jakarta EE dependencies with a `jakarta.` group ID in Spring Boot's starters. Dependency management for the Jakarta EE API dependencies has been added alongside the existing dependency management for the Java EE API dependencies. The dependency management for the Java EE API dependencies will be removed in the future and all users are encouraged to move to the Jakarta EE API dependencies.

As part of the Java EE to Jakarta EE migration, two dependencies have changed artifact ID in their latest maintenance releases. `com.sun.mail:javax.mail` is now `com.sun.mail:jakarta.mail` and `org.glassfish:javax.el` is now `org.glassfish:jakarta.el`. In the unlikely event that you were using either of these dependencies directly, please update your `pom.xml` or `build.gradle` accordingly.

JUnit 5

`spring-boot-starter-test` now provides JUnit 5 by default. JUnit 5's vintage engine is included by default to support existing JUnit 4-based test classes so that you can migrate to JUnit 5 when you are ready to do so. It is also possible to use a mixture of JUnit 4- and JUnit 5-based test classes in the same module. This allows you to migrate to JUnit 5 gradually if you wish.

Note that the Maven Surefire plugin for JUnit 4 does not support the `listener` property. If you have a Maven configuration similar to the following:

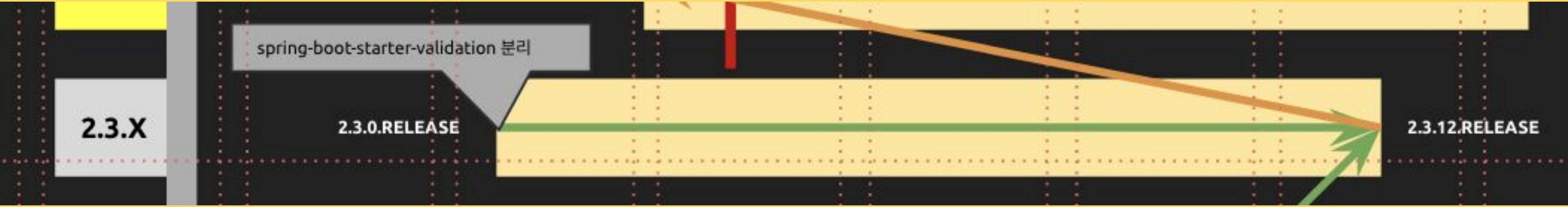
```
<configuration>
        <properties>
                <property>
                        <name>listener</name>
                        <value>com.example.CustomRunListener</value>
                </property>
        </properties>
</configuration>
```

You can't use the `junit-vintage-engine` and you'll need to explicitly roll back to JUnit 4:

JUnit Migration: JUnit4 -> JUnit5

- Spock 안녕!

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.2-Release-Notes



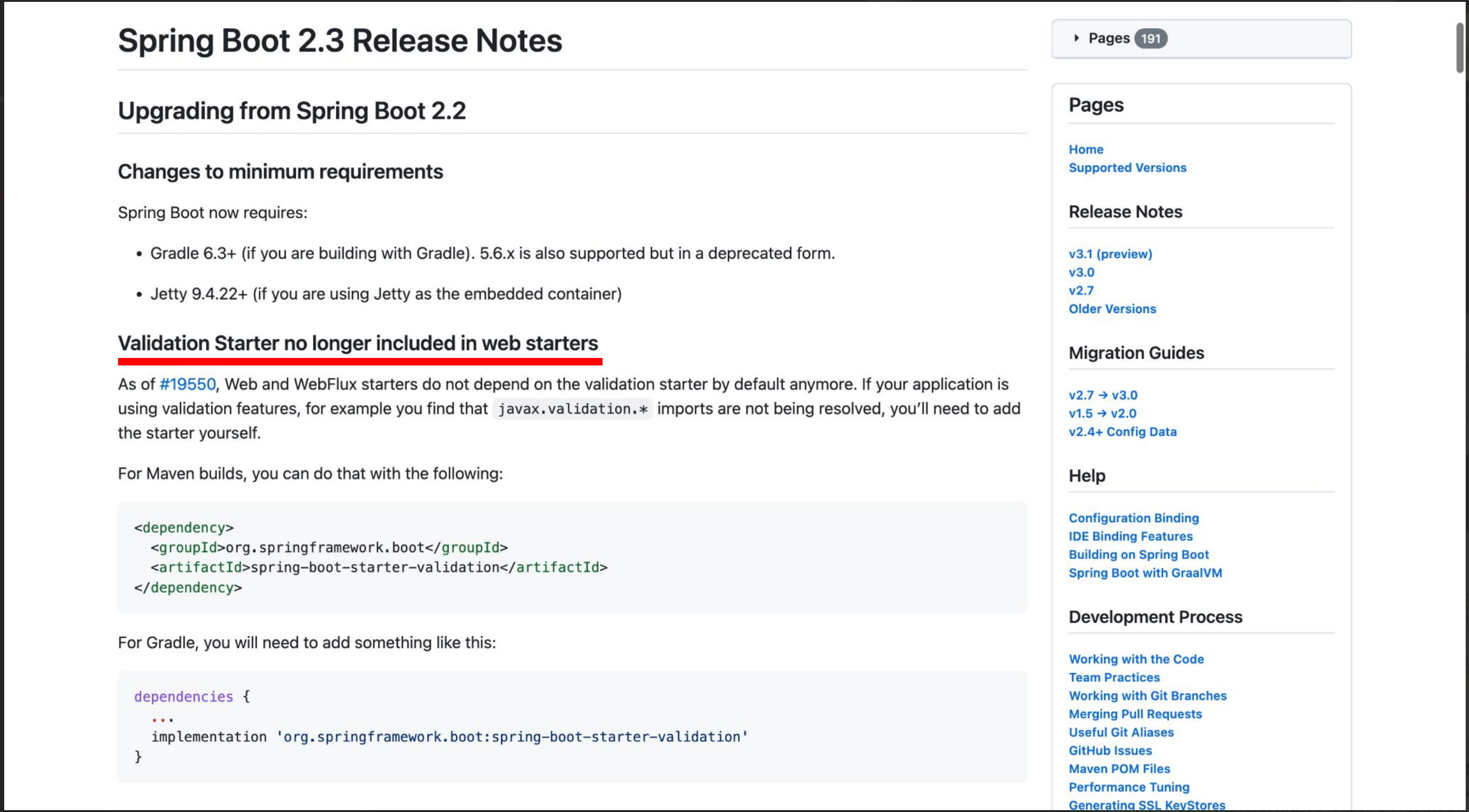
2.3.X

spring-boot-starter-validation 분리

2.3.0.RELEASE

2.3.12.RELEASE

spring-boot-starter:2.3.X.RELEASE



Spring Boot 2.3 Release Notes

Upgrading from Spring Boot 2.2

Changes to minimum requirements

Spring Boot now requires:

- Gradle 6.3+ (if you are building with Gradle). 5.6.x is also supported but in a deprecated form.
- Jetty 9.4.22+ (if you are using Jetty as the embedded container)

Validation Starter no longer included in web starters

As of #19550, Web and WebFlux starters do not depend on the validation starter by default anymore. If your application is using validation features, for example you find that `javax.validation.*` imports are not being resolved, you'll need to add the starter yourself.

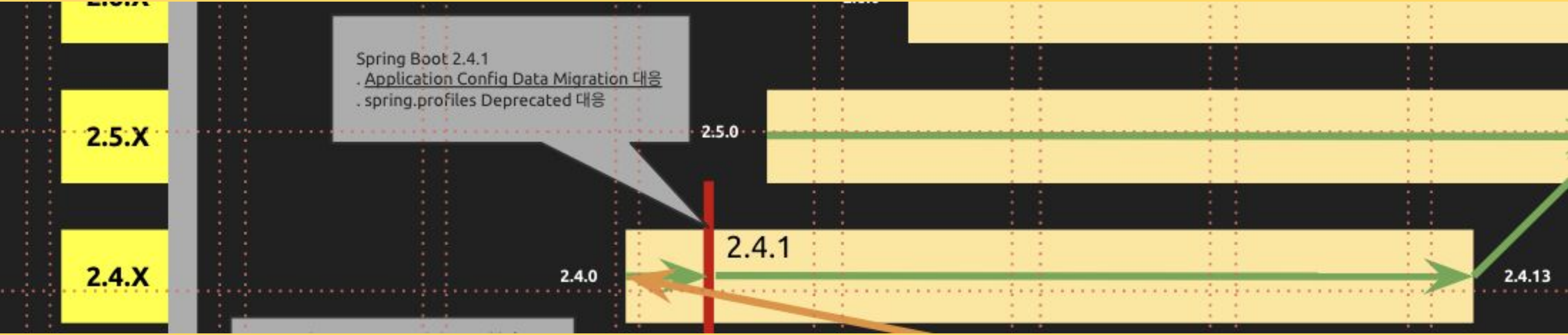
For Maven builds, you can do that with the following:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
```

For Gradle, you will need to add something like this:

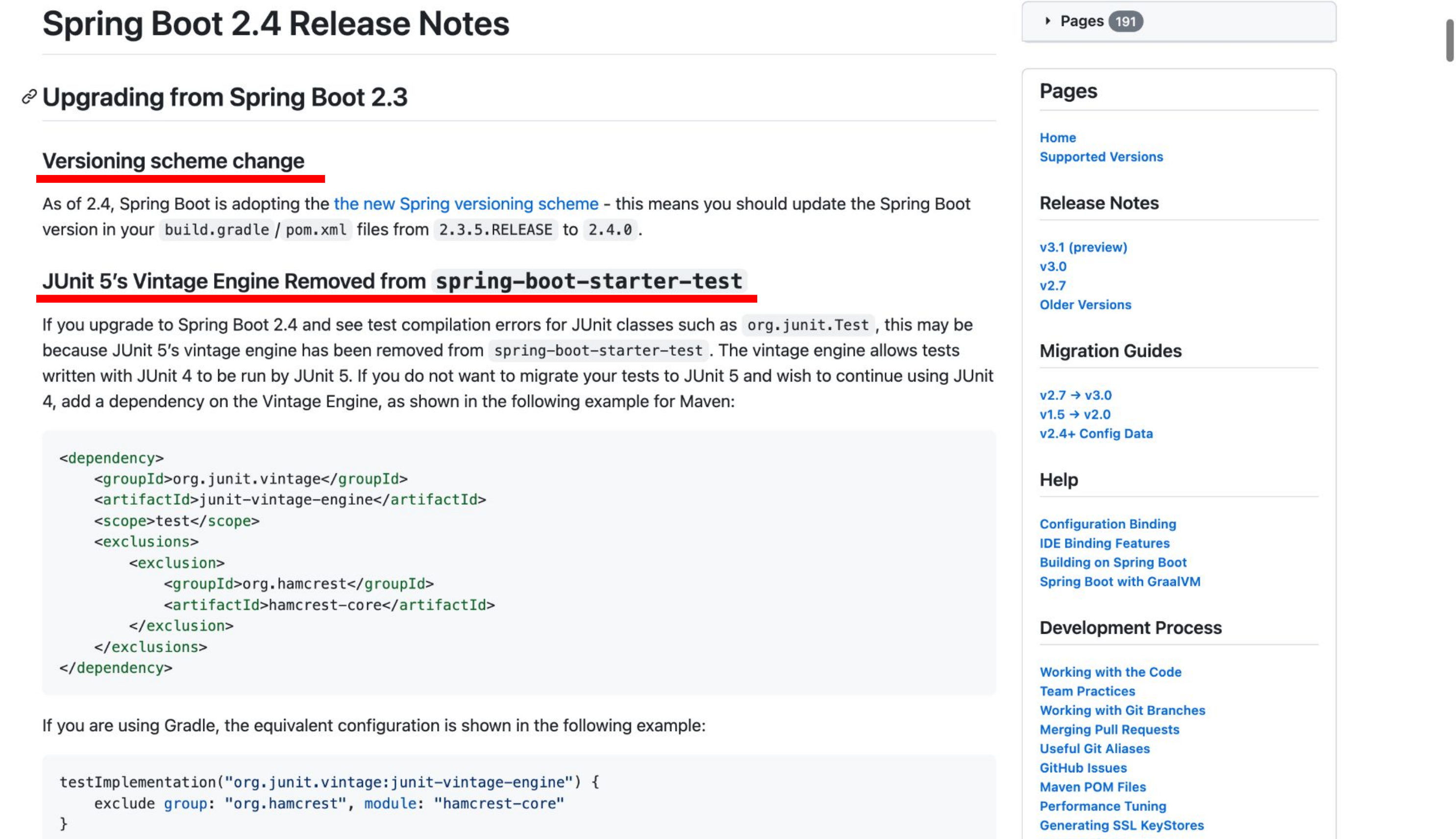
```
dependencies {
  ...
  implementation 'org.springframework.boot:spring-boot-starter-validation'
}
```

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.3-Release-Notes



Spring Boot 2.4.1
. Application Config Data Migration 대응
. spring.profiles Deprecated 대응

spring-boot-starter:2.4.X.



Spring Boot 2.4 Release Notes

Upgrading from Spring Boot 2.3

Versioning scheme change

As of 2.4, Spring Boot is adopting the the new Spring versioning scheme - this means you should update the Spring Boot version in your `build.gradle` / `pom.xml` files from `2.3.5.RELEASE` to `2.4.0`.

JUnit 5's Vintage Engine Removed from `spring-boot-starter-test`

If you upgrade to Spring Boot 2.4 and see test compilation errors for JUnit classes such as `org.junit.Test`, this may be because JUnit 5's vintage engine has been removed from `spring-boot-starter-test`. The vintage engine allows tests written with JUnit 4 to be run by JUnit 5. If you do not want to migrate your tests to JUnit 5 and wish to continue using JUnit 4, add a dependency on the Vintage Engine, as shown in the following example for Maven:

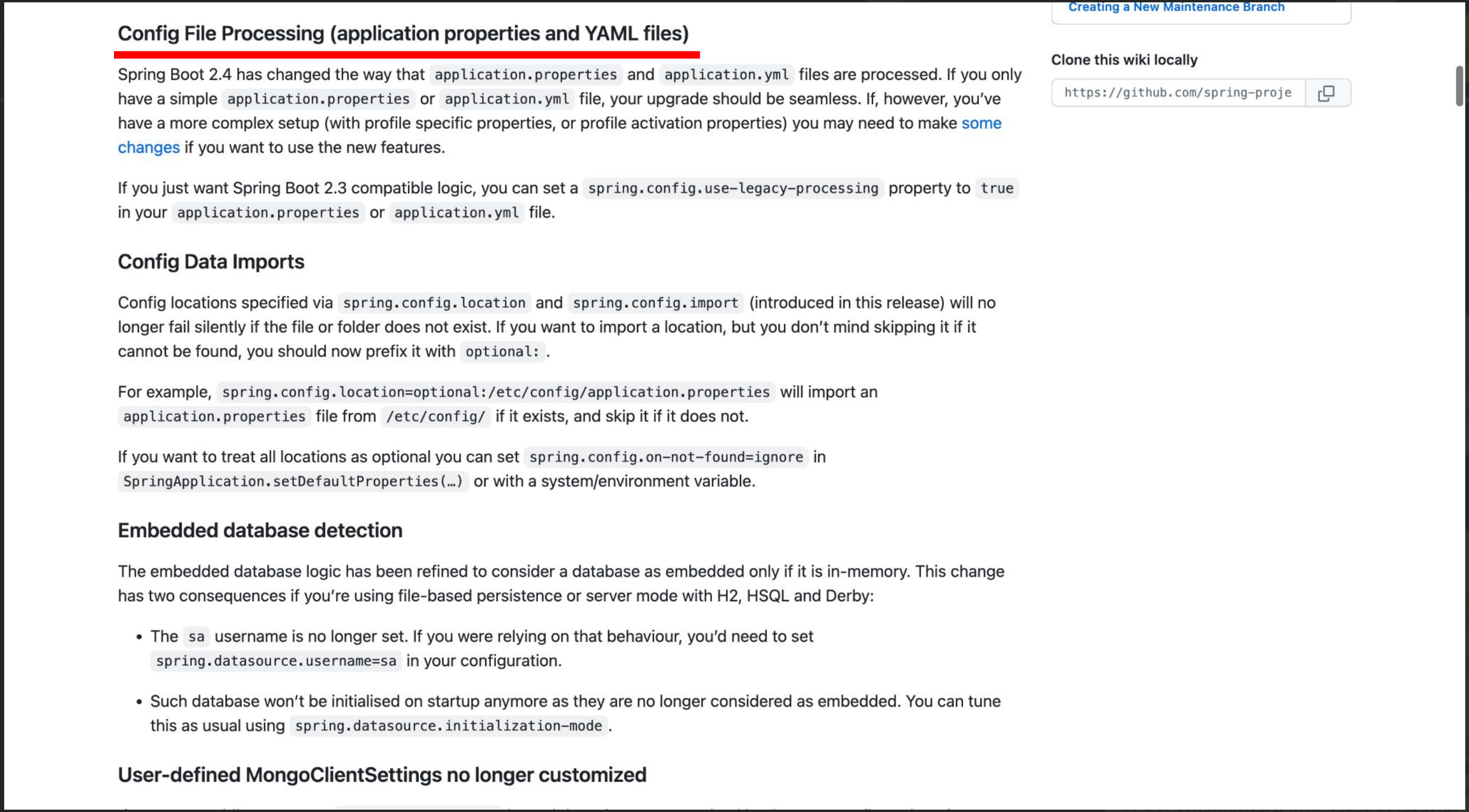
```
<dependency>
    <groupId>org.junit.vintage</groupId>
    <artifactId>junit-vintage-engine</artifactId>
    <scope>test</scope>
    <exclusions>
        <exclusion>
            <groupId>org.hamcrest</groupId>
            <artifactId>hamcrest-core</artifactId>
        </exclusion>
    </exclusions>
</dependency>
```

If you are using Gradle, the equivalent configuration is shown in the following example:

```
testImplementation("org.junit.vintage:junit-vintage-engine") {
    exclude group: "org.hamcrest", module: "hamcrest-core"
}
```

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.4-Release-Notes

spring-boot-starter:2.4.X.


Creating a New Maintenance Branch
Clone this wiki locally
https://github.com/spring-proje

Config File Processing (application properties and YAML files)

Spring Boot 2.4 has changed the way that `application.properties` and `application.yml` files are processed. If you only have a simple `application.properties` or `application.yml` file, your upgrade should be seamless. If, however, you've have a more complex setup (with profile specific properties, or profile activation properties) you may need to make some changes if you want to use the new features.

If you just want Spring Boot 2.3 compatible logic, you can set a `spring.config.use-legacy-processing` property to `true` in your `application.properties` or `application.yml` file.

Config Data Imports

Config locations specified via `spring.config.location` and `spring.config.import` (introduced in this release) will no longer fail silently if the file or folder does not exist. If you want to import a location, but you don't mind skipping it if it cannot be found, you should now prefix it with `optional:`.

For example, `spring.config.location=optional:/etc/config/application.properties` will import an `application.properties` file from `/etc/config/` if it exists, and skip it if it does not.

If you want to treat all locations as optional you can set `spring.config.on-not-found=ignore` in `SpringApplication.setDefaultProperties(…)` or with a system/environment variable.

Embedded database detection

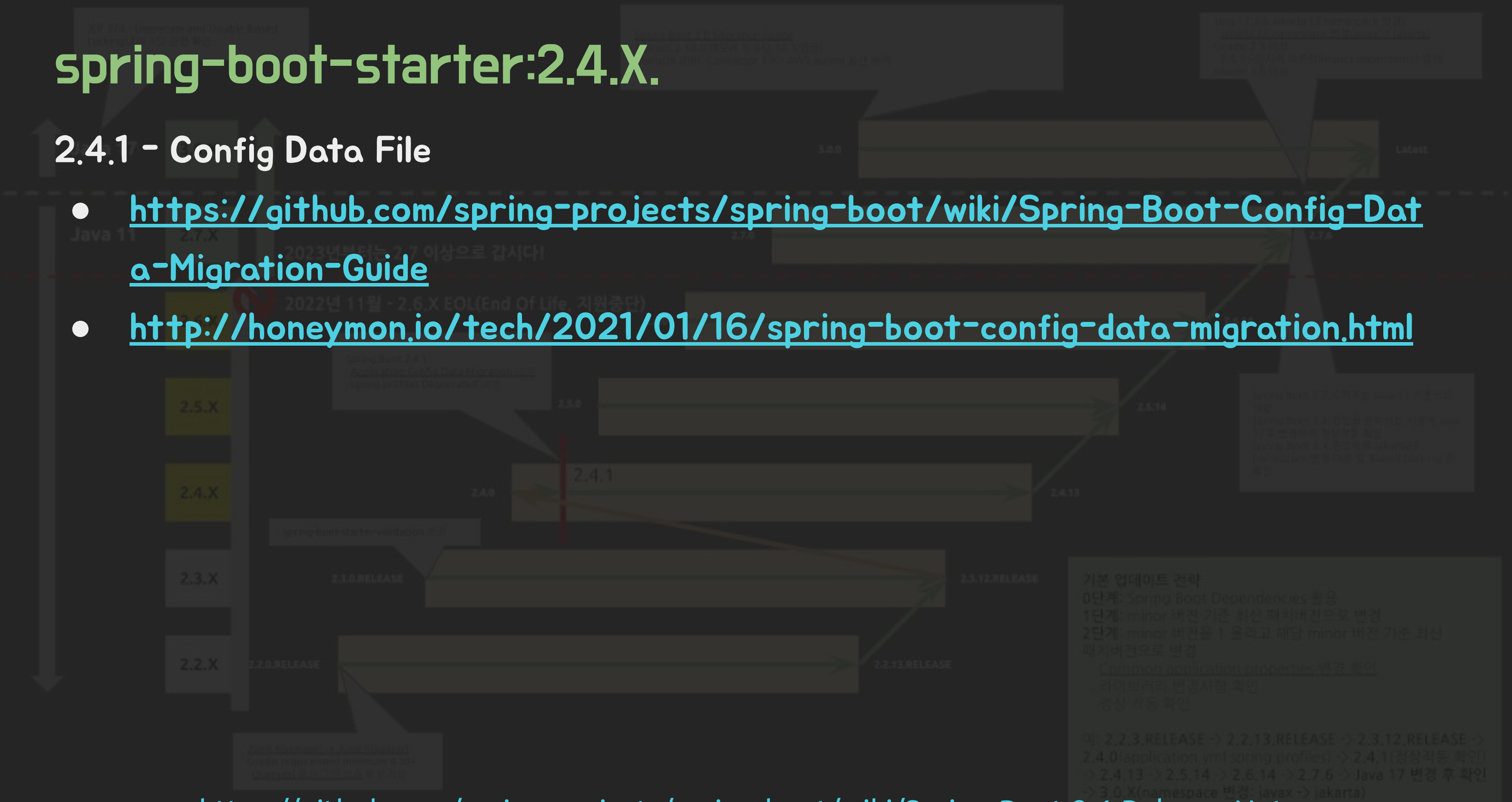
The embedded database logic has been refined to consider a database as embedded only if it is in-memory. This change has two consequences if you're using file-based persistence or server mode with H2, HSQL and Derby:

- The `sa` username is no longer set. If you were relying on that behaviour, you'd need to set `spring.datasource.username=sa` in your configuration.
- Such database won't be initialised on startup anymore as they are no longer considered as embedded. You can tune this as usual using `spring.datasource.initialization-mode`.

User-defined MongoClientSettings no longer customized

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.4-Release-Notes

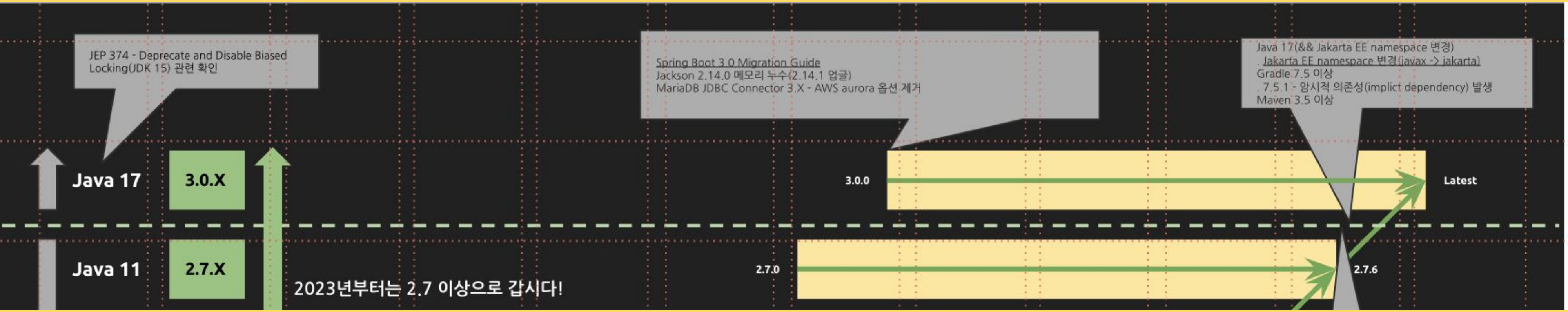
spring-boot-starter:2.4.X.


Java 11
2.7.X
2023년부터는 2.7 이상으로 갑시다!
2022년 11월 - 2.6.X EOL(End Of Life, 지원중단)
2.5.X
2.4.X
2.4.1
2.3.X
2.2.X
기본 업데이트 전략

2.4.1 - Config Data File

- https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-Config-Data-Migration-Guide
- http://honeymon.io/tech/2021/01/16/spring-boot-config-data-migration.html

https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.4-Release-Notes



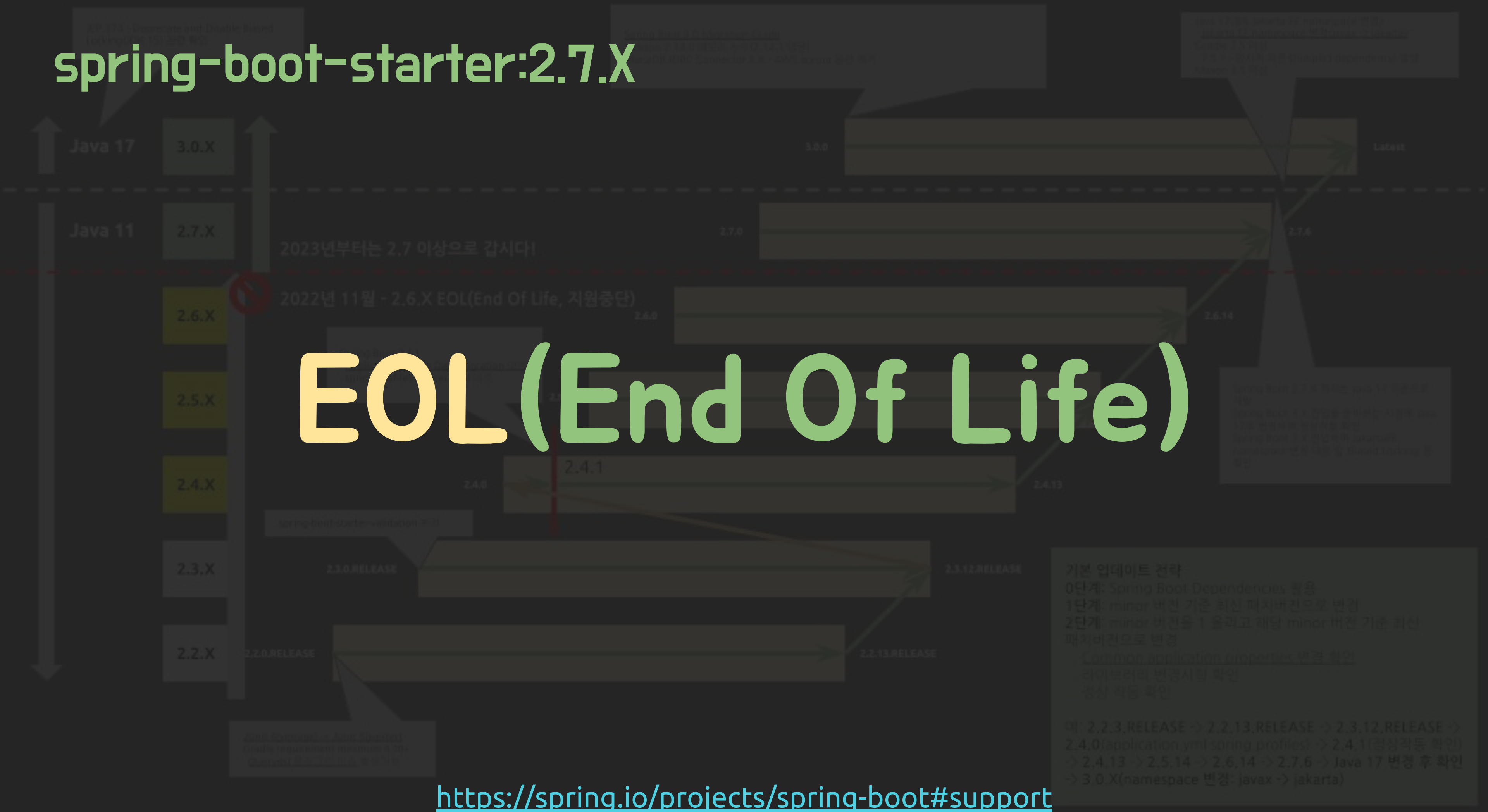
JEP 374 - Deprecate and Disable Biased Locking(JDK 15) 관련 확인

Spring Boot 3.0 Migration Guide
Jackson 2.14.0 메모리 누수(2.14.1 업글)
MariaDB JDBC Connector 3.X - AWS aurora 옵션 제거

Java 17(&& Jakarta EE namespace 변경)
. Jakarta EE namespace 변경(javax -> jakarta)
Gradle 7.5 이상
. 7.5.1 - 암시적 의존성(implict dependency) 발생
Maven 3.5 이상

2023년부터는 2.7 이상으로 갑시다!

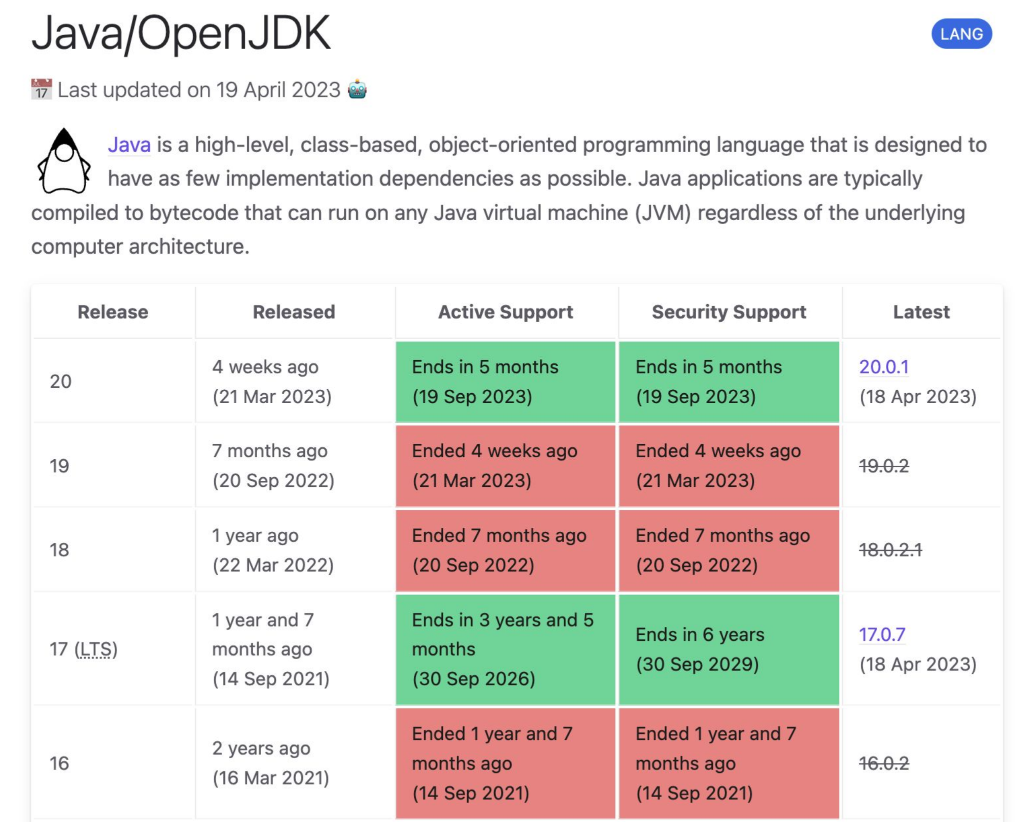
spring-boot-starter:2.7.X


Java 17
3.0.X
Java 11
2.7.X
2023년부터는 2.7 이상으로 갑시다!
2.6.X
2022년 11월 - 2.6.X EOL(End Of Life, 지원중단)
2.5.X
2.4.X
2.4.1
2.3.X
2.2.X
2.2.0.RELEASE
2.2.13.RELEASE
2.3.0.RELEASE
2.3.12.RELEASE
2.4.0
2.4.13
2.6.0
2.6.14
2.7.0
2.7.6
3.0.0
Latest
기본 업데이트 전략
0단계: Spring Boot Dependencies 등록
1단계: minor 버전 기준 최신 패치버전으로 변경
2단계: minor 버전을 1 올리고 해당 minor 버전 기준 최신 패치버전으로 변경
Common application properties 변경 확인
라이브러리 변경사항 확인
정상 작동 확인
예: 2.2.3.RELEASE -> 2.2.13.RELEASE -> 2.3.12.RELEASE -> 2.4.0(application.yml spring.profiles) -> 2.4.1(정상작동 확인) -> 2.4.13 -> 2.5.14 -> 2.6.14 -> 2.7.6 -> Java 17 변경 후 확인 -> 3.0.X(namespace 변경: javax -> jakarta)

EOL(End Of Life)

https://spring.io/projects/spring-boot#support

spring-boot-starter:2.7.X



Java/OpenJDK

LANG

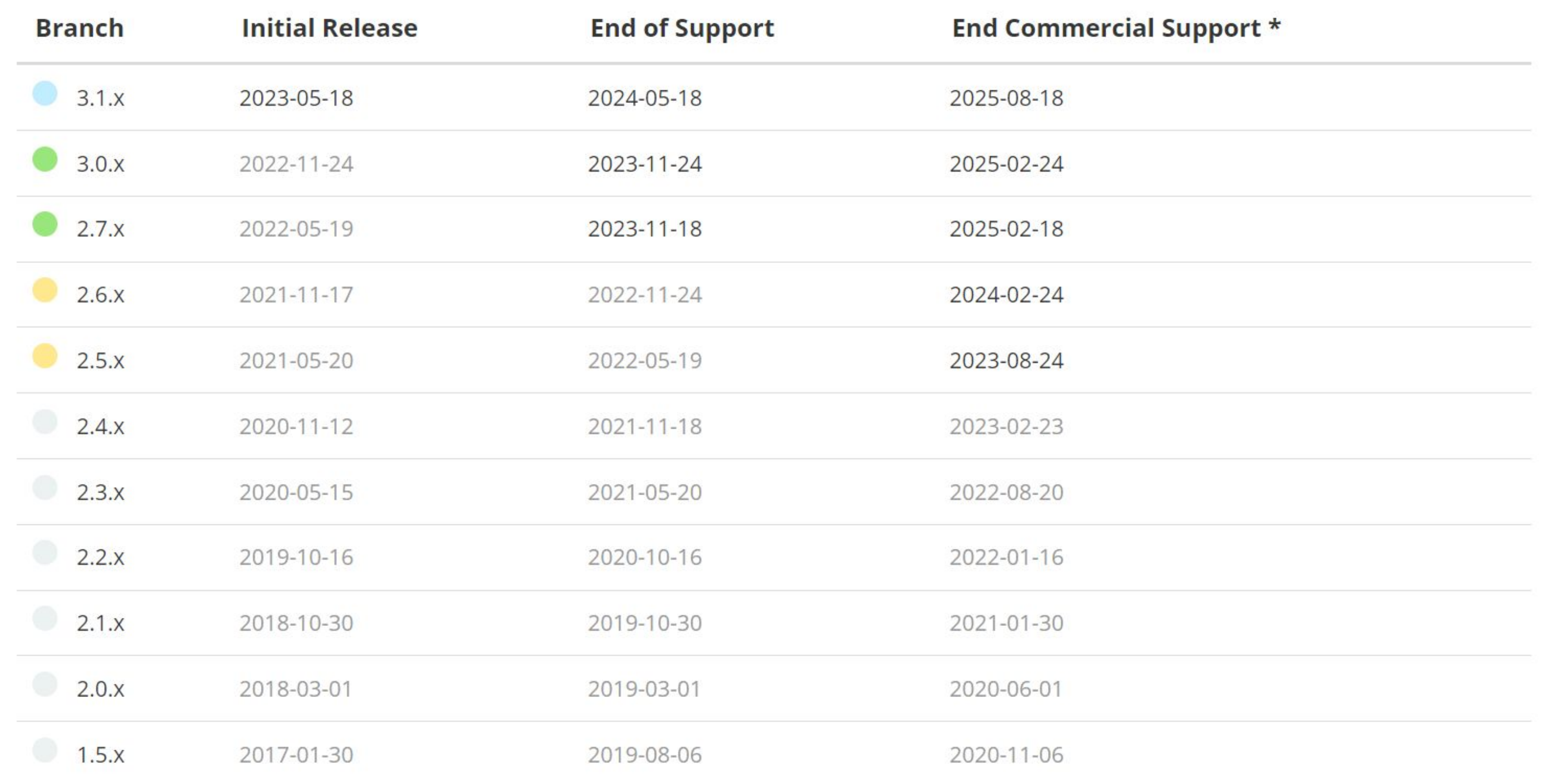
📅 Last updated on 19 April 2023 🤖

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

Release	Released	Active Support	Security Support	Latest
20	4 weeks ago (21 Mar 2023)	Ends in 5 months (19 Sep 2023)	Ends in 5 months (19 Sep 2023)	20.0.1 (18 Apr 2023)
19	7 months ago (20 Sep 2022)	Ended 4 weeks ago (21 Mar 2023)	Ended 4 weeks ago (21 Mar 2023)	~~19.0.2~~
18	1 year ago (22 Mar 2022)	Ended 7 months ago (20 Sep 2022)	Ended 7 months ago (20 Sep 2022)	~~18.0.2.1~~
17 (LTS)	1 year and 7 months ago (14 Sep 2021)	Ends in 3 years and 5 months (30 Sep 2026)	Ends in 6 years (30 Sep 2029)	17.0.7 (18 Apr 2023)
16	2 years ago (16 Mar 2021)	Ended 1 year and 7 months ago (14 Sep 2021)	Ended 1 year and 7 months ago (14 Sep 2021)	~~16.0.2~~

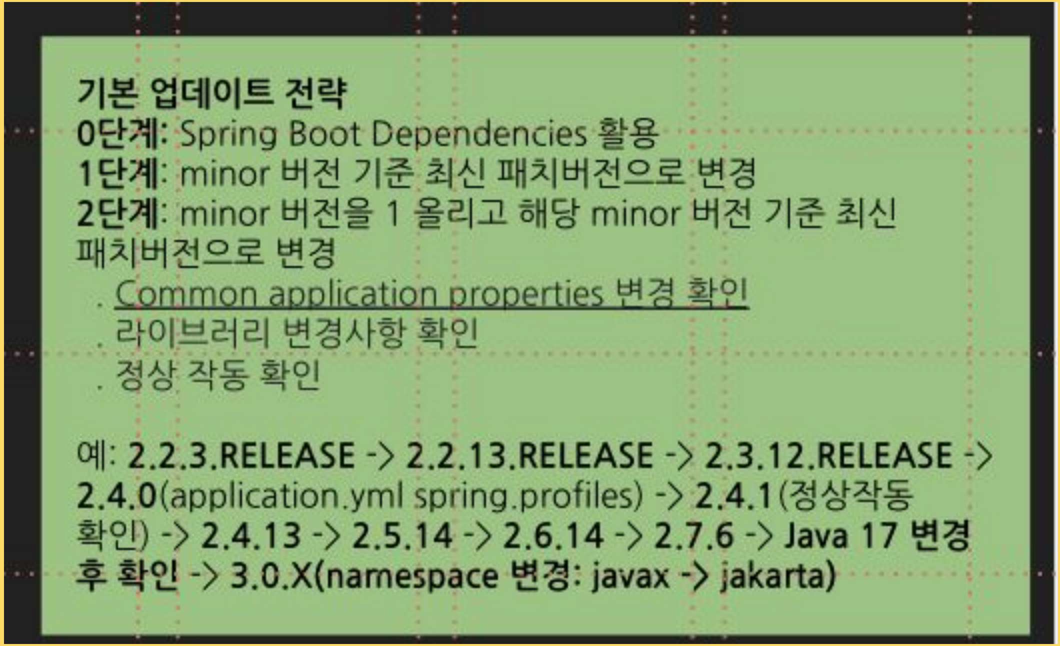
https://endoflife.date/java

spring-boot-starter:2.7.X



Branch	Initial Release	End of Support	End Commercial Support *
🔵 3.1.x	2023-05-18	2024-05-18	2025-08-18
🟢 3.0.x	2022-11-24	2023-11-24	2025-02-24
🟢 2.7.x	2022-05-19	2023-11-18	2025-02-18
🟡 2.6.x	2021-11-17	2022-11-24	2024-02-24
🟡 2.5.x	2021-05-20	2022-05-19	2023-08-24
⚪ 2.4.x	2020-11-12	2021-11-18	2023-02-23
⚪ 2.3.x	2020-05-15	2021-05-20	2022-08-20
⚪ 2.2.x	2019-10-16	2020-10-16	2022-01-16
⚪ 2.1.x	2018-10-30	2019-10-30	2021-01-30
⚪ 2.0.x	2018-03-01	2019-03-01	2020-06-01
⚪ 1.5.x	2017-01-30	2019-08-06	2020-11-06

https://spring.io/projects/spring-boot#support



기본 업데이트 전략

0단계: Spring Boot Dependencies 활용
1단계: minor 버전 기준 최신 패치버전으로 변경
2단계: minor 버전을 1 올리고 해당 minor 버전 기준 최신 패치버전으로 변경

. Common application properties 변경 확인
. 라이브러리 변경사항 확인
. 정상 작동 확인

예: **2.2.3.RELEASE -> 2.2.13.RELEASE -> 2.3.12.RELEASE -> 2.4.0**(application.yml spring.profiles) -> **2.4.1**(정상작동 확인) -> **2.4.13 -> 2.5.14 -> 2.6.14 -> 2.7.6 -> Java 17 변경 후 확인** -> **3.0.X(namespace 변경: javax -> jakarta)**

스프링 부트 업그레이 전략!

기본 업데이트 전략

- 0단계: Spring Boot Dependencies 활용
- 1단계: minor 버전 기준 최신 패치버전으로 변경
- 2단계: minor 버전을 1 올리고 해당 minor 버전 기준 최신 패치버전으로 변경
  - Common application properties 변경 확인
  - 라이브러리 변경사항 확인
  - 정상 작동 확인
- 업그레이드 버전 전략:

예: 2.2.3.RELEASE -> 2.2.13.RELEASE -> 2.3.12.RELEASE -> 2.4.0(application.yml spring.profiles) -> 2.4.1(정상작동 확인) -> 2.4.13 -> 2.5.14 -> 2.6.14 -> 2.7.6 -> Java 17 변경 후 확인 -> 3.0.X(namespace 변경: javax -> jakarta)

Career Excellence

with Kurly

**컬리에서 '진정성 있는 도전'을
함께 할 여러분을 기다립니다.**

컬리는 기술과 데이터를 통해 유통을 혁신하며,
모두를 위한 더 나은 삶을 만들기 위해 존재해요.

https://kurly.career.greetinghr.com/

진짜 끝!

내년에는 새로운 발표자들이 많아지길!!

궁금하신 것은 ihoneymon@gmail.com 으로 문의주세요.

참조

- https://spring.io/
  - https://spring.io/blog
- https://github.com/spring-projects/spring-boot
- https://github.com/spring-projects/spring-framework
- 인프런: https://www.inflearn.com/
  - 토비님: https://www.inflearn.com/users/@tobyilee
  - 김영한님: https://www.inflearn.com/users/@yh
- https://semver.org/lang/ko/
- Reference Documetation
  - Spring Framework: https://docs.spring.io/spring-framework/docs/current/reference/html/index.html
  - Spring Boot: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

참조

Java

- https://docs.oracle.com/en/java/javase/17/index.html

Jakarta

- https://jakarta.ee/release/10/
- https://eclipse-ee4j.github.io/jakartaee-platform/jakartaee9/JakartaEE9ReleasePlan
- https://eclipse-ee4j.github.io/jakartaee-platform/jakartaee10/JakartaEE10ReleasePlan

Jakarta Persistence & Hibernate

- https://hibernate.org/orm/releases/
- https://github.com/jakartaee/persistence/releases/tag/3.1-3.1.0-RELEASE
- https://newsroom.eclipse.org/eclipse-newsletter/2022/march/what%E2%80%99s-new-jakarta-persistence-31
- https://vladmihalcea.com/uuid-database-primary-key/
- https://github.com/vladmihalcea/hypersistence-tsid

Jackson

- https://blogs.oracle.com/javamagazine/post/java-json-serialization-jackson
- https://dev.to/brunooliveira/practical-java-16-using-jackson-to-serialize-records-4og4

참고

연관자료

- 어느 월급쟁이개발자의 스프링 부트 따라잡기: https://www.youtube.com/watch?v=cPMG9MJqCnQ&t=14s
  - 발표장표: https://docs.google.com/presentation/d/1hDA6cQPceFu5flJfePc_4SFccH_XWZME5olNxx1K6uE/edit?usp=sharing
- 스프링 부트 3 살펴보기
  - 장표: https://docs.google.com/presentation/d/1Vezn8gZXDdb7l10eDMHR_j0e04ip0D0Buju2lXcf07o/edit?usp=sharing