

**초격차 패키지 Online.**

# 버전별 변천사

## **PART1** | 버전별 변천사 훑어보기: 1 vs 2

무엇이 달라졌나요?

## **PART2** | 버전별 변천사 훑어보기: 2.1 ~ 2.3

2.1, 2.2, 2.3 지나온 길 돌아보기

## **PART3** | 버전별 변천사 훑어보기: 2.4

2.4 에서 더 좋아진 것들

## **PART4** | 버전별 변천사 훑어보기: 2.5

2.5 에서 더 좋아진 것들

# 버전별 변천사

**2** 버전별 변천사 훑어보기: 2.1 ~ 2.3

# Spring Boot 2.1

## Spring Boot 2.1

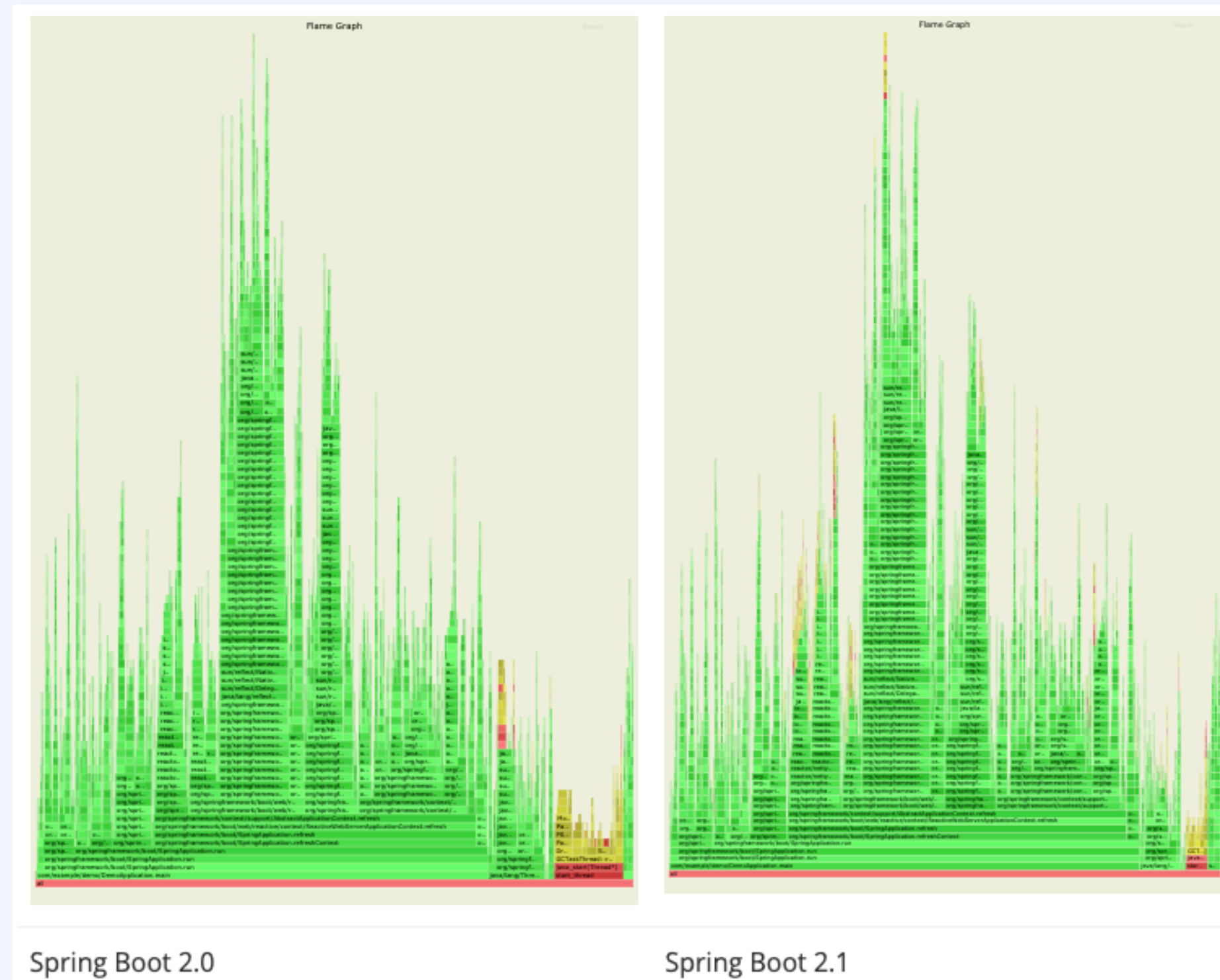
**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 성능 향상
- Spring Data JPA: bootstrap-mode
- JDK 11 + Spring Framework 5.1
- 주요 버전 업그레이드
- Deprecations from 2.0
- New Spring Actuator Endpoints
- Logging Refinements
- JUnit
- Bean Overriding

## 성능 향상

2.  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- <https://spring.io/blog/2018/12/12/how-fast-is-spring>
- 더 빨라짐!
  - GC 속도 향상 - 더 빠른 시작
  - 더 적은 heap 사용



## Spring Data JPA: bootstrap-mode

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Spring Data Lovelace
- lazy initialization
- `spring.data.jpa.repositories.bootstrap-mode=deferred(or lazy)`

## JDK 11 + Spring Framework 5.1

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Java 11
- Spring Framework 5.1

## 주요 버전 업그레이드

- Tomcat 9
- Undertow 2
- Hibernate 5.3
- JUnit 5.2
- Micrometer 1.1

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3



## Deprecations from 2.0

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- `setConnectTimeout(int)`, `setReadTimeout(int)`
- data size 를 int 입력 인자로 받던 메소드들은 Datasize 로 변경 및 deprecated
- `management.endpoints.jmx.unique-names` -> `spring.jmx.unique-names`
- `@AutoConfigureMockMvc` 의 `secure` 속성 -> `@WithMockUser`
- `RestTemplateBuilder.basicAuthorization` -> `basicAuthentication`
- `AbstractFilterRegistrationBean`, `FilterRegistrationBean`
  - as-is: `REQUEST_WRAPPER_FILTER_MAX_ORDER`
  - to-be: `OrderedFilter.REQUEST_WRAPPER_FILTER_MAX_ORDER`

## New Spring Actuator Endpoints

**2.**  
버전별 변경사항  
확인하기: 2.1 ~ 2.3

- /actuator/caches 추가
- /actuator/integrationgraph 추가
- /actuator/health 기능 업데이트
  - HealthIndicator 를 런타임에 추가, 삭제 가능
  - 특정 인디케이터만 관찰 가능 (ex: /actuator/health/db)

## Logging Refinements

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- `logging.level.web=debug`
- `spring.http.log-request-details=true`

## JUnit

**2.**  
버전별 변환사  
훑어보기: 2.1 ~ 2.3

- JUnit 5 사용 가능
  - spring-boot-test -> 5.3.1
- 그러나 기본 세팅은 JUnit 4
- JUnit 5로 바꾸려면 빌드 설정에 명시해야 함
- @ExtendWith(SpringExtension.class) 을 모든 부트 @-Test 애노테이션에 내재화

## Bean Overriding

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 사용자가 모르게 빈이 오버라이드 되지 않도록, 기본 동작 false
  - 오버라이드가 감지되면: BeanDefinitionOverrideException
- `spring.main.allow-bean-definition-overriding=true`

# Spring Boot 2.2

## Spring Boot 2.2

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 성능 향상
- Java 13 + Spring Framework 5.2
- 주요 버전 업그레이드
- Deprecations from 2.1
- Jakarta EE
- Junit 5
- Gradle 4.10

## 성능 향상

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 시작 시간과 메모리 사용이 더욱 감소
- 새 속성 추가: proxyBeanMethods=false
  - @Configuration
  - @SpringBootApplication
  - @SpringBootConfiguration
- spring-boot:run, bootRun -> -Xverify:none -XX:TieredStopAtLevel=1
  - JDK 13 쓸 경우: -Xverify:none 가 deprecated 되었기 때문에 빠짐
- spring.main.lazy-initialization
  - 부작용 1: http request 핸들링 속도가 조금 더 길어짐
  - 부작용 2: 시작할 때 문제가 있다면 보여주던 실패를 더이상 보여주지 않음



## Java 13 + Spring Framework 5.2

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Java 13 지원 추가
- Spring Framework 5.2

## 주요 버전 업그레이드

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Spring HATEOAS 1.0
- Spring Kafka 2.3
- Spring Security 5.2
- Hamcrest 2.1
- Mockito 3.1
- AssertJ 3.12
- Elasticsearch 6.7
- Git Commit ID Plugin 3.0
- Hazelcast 3.12
- Jackson 2.10
- Jedis 3.1
- Lettuce 5.2

## Deprecations from 2.1

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 스프링 부트 2.1 에서 deprecated 된 클래스, 메소드, 프로퍼티들이 삭제됨
- <https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.1-Release-Notes#deprecations-in-spring-boot-21>

## Jakarta EE

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- javax. group ID -> jakarta. 로 이동
- com.sun.mail:javax.mail -> com.sun.mail:jakarta.mail
- org.glassfish:javax.el -> org.glassfish:jakarta.el

## JUnit 5

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 드디어 JUnit 5 기본 제공

## Gradle 4.10

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 버전의 최소 조건이 4.10 으로 변경됨

# Spring Boot 2.3

## Spring Boot 2.3

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 새 버전의 배포 주기 계획 발표
- Java 14, Gradle 5.6.x ~ 6.3, developmentOnly Gradle 설정 자동 생성
- 주요 버전 업그레이드 (R2DBC 지원)
- 도커 지원
- Validation Starter 를 web starter 에서 제외
- Graceful shutdown
- Jackson 2.11
- Actuator 향상점
- 기본 에러 페이지 내용 변화
- @ActiveProfiles 에 여러 개의 프로파일 설정하기



## 새 버전의 배포 주기 계획 발표

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 앞으로 스프링 부트 새 버전은 6개월 주기로 배포하기로 함
- 배포를 예상 가능, 업그레이드 주기를 계획할 수 있음
- 실제로
  - 2.3: 2020. 5. 15
  - 2.4: 2020. 11. 12
  - 2.5: 2021. 5. 25
  - 2.6: 2021.11. ?? (2.6.0.SNAPSHOT 현재 이용 가능)
- Spring Framework 는 feature-based 배포 전략을 유지 "준비되면 나가겠다"

## Java 14, Gradle 5.6.x ~ 6.3, developmentOnly Gradle 설정 자동 생성

2.

버전별 변천사

훑어보기: 2.1 ~ 2.3

- Java 14 지원, 8과 11도 지원
- Gradle 6.3 필요 (5.6.x 은 지원하지만 deprecated 형태)
- developmentOnly 가 스프링 부트 플러그인에 의해 자동 생성됨
  - developmentOnly 에 추가 설정을 수동으로 구성한 코드는 모두 삭제해야 함
- 그러지 않을 경우: "cannot add a configuration with name 'developmentOnly' as a configuration with that name already exists."

## 주요 버전 업그레이드 (R2DBC 지원)

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Spring Data Neumann -> R2DBC 지원
  - R2DBC ConnectionFactory 자동 설정, health indicator, metrics, @DataR2dbcTest
- Spring HATEOAS 1.1
- Spring Integration 5.3
- Spring Kafka 2.5
- Spring Security 5.3
- AssertJ 3.16, JUnit Jupiter 5.6, Mockito 3.3
- Elasticsearch 7.6
- Hibernate Validator 6.1
- Jackson 2.11
- QueryDSL 4.3
- MongoDB 4.0

## 도커 지원

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- Cloud Native Buildpacks
  - Maven: spring-boot:build-image
  - Gradle: bootBuildImage
  - Default Java buildpack: Paketo
- layered jars
  - jar 내용물을 나눠서, 업데이트가 있는 이미지만 빌드하는 방식
  - 도커 이미지를 더 효율적으로 작성하는데 도움

## Validation Starter 를 web starter 에서 제외

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- javax.validation.\* 이 기본으로 포함되지 않게 됨
  - ex: @Valid, @NotNull, @Min, @Max, @Email, ...
- 이유: 검증 기능이 모든 서비스에서 필요한 것은 아닌데, 디펜던시가 부피를 꽤 차지함
- implementation 'org.springframework.boot:spring-boot-starter-validation'

## Graceful shutdown

2.

버전별 변천사

확인하기: 2.1 ~ 2.3

- 4가지 임베디드 웹 서버에 이 기능을 제공
  - Jetty, Reactor Netty, Tomcat, Undertow
- `server.shutdown=graceful`
- shutdown 실행되면, 바로 죽지 않고 모든 새로운 요청을 거부, 남은 요청들이 처리되기를 기다림
- `spring.lifecycle.timeout-per-shutdown-phase`: grace period 설정
  - 남은 요청이 처리되기를 기다리는 시간 (default: 30s)

## Jackson 2.11

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- java.util.Date, java.util.Calendar 기본 포매팅에 변화
- 의도: 정확한 ISO-8601 확장 규격을 준수
  - as-is: "2021-01-02T03:04:56.789+0000"
  - to-be: "2021-01-02T03:04:56.789+00:00"

## Actuator 향상점

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- /actuator/configprops 가 end-to-end 정보를 추적하여 제공
- /actuator/metrics: 이름들이 알파벳 순 보기좋게 정렬됨
- 기타 등등



## 기본 에러 페이지 내용 변화

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3

- 에러 메시지, 바인딩 에러가 기본 에러 페이지에 노출되지 않음
- 클라이언트에 정보 유출을 방지
- `server.error.include-message`
- `server.error.include-binding-errors`

## @ActiveProfiles 에 여러 개의 프로파일 설정하기

- @ActiveProfiles({"p1","p2"})
- 이제 이러면 안됨: @ActiveProfiles("p1,p2")

**2.**  
버전별 변천사  
훑어보기: 2.1 ~ 2.3