

Chapter 04. Spring Boot

# Spring Boot 시작하기

## Chapter 04. Spring Boot

# Spring Boot

## § 8. Introducing Spring Boot

Spring Boot helps you to create stand-alone, production-grade Spring-based Applications that you can run. We take an opinionated view of the Spring platform and third-party libraries, so that you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

You can use Spring Boot to create Java applications that can be started by using `java -jar` or more traditional war deployments. We also provide a command line tool that runs “spring scripts”.

Our primary goals are:

- Provide a radically faster and widely accessible getting-started experience for all Spring development.
- Be opinionated out of the box but get out of the way quickly as requirements start to diverge from the defaults.
- Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).
- Absolutely no code generation and no requirement for XML configuration.

<https://docs.spring.io/spring-boot/docs/2.1.x/reference/html/getting-started-introducing-spring-boot.html>

- Spring Boot 는 단순히 실행되며, 프로덕션 제품 수준의 스프링 기반 어플리케이션을 쉽게 만들 수 있다.
- Spring Boot 어플리케이션에는 **Spring 구성**이 거의 필요 하지 않다.
- Spring Boot **java -jar** 로 실행하는 **Java 어플리케이션**을 만들 수 있다.

#### 주요 목표

- Spring 개발에 대해 빠르고, 광범위하게 적용할 수 있는 환경
- 기본값 설정이 있지만 설정을 바꿀 수 있다.
- 대규모 프로젝트에 공통적인 비 기능 제공 ( 보안, 모니터링 등등)
- XML 구성 요구사항이 전혀 없음

## Build Tool

Name	version
Maven	3.3+
Gradle	4.x (4.4 and later) and 5.x

## Servlet Containers

Name	servlet version
Tomcat 9.x	3.3
Jetty 9.4	3.1
Undertow 2.0	4.0
Netty	-

<https://start.spring.io>



Project

☐ Maven Project ☒ Gradle Project

Language

☐ Java ☒ Kotlin ☐ Groovy

Spring Boot

☐ 2.4.0 (SNAPSHOT) ☐ 2.4.0 (M2) ☐ 2.3.4 (SNAPSHOT) ☒ 2.3.3  
☐ 2.2.10 (SNAPSHOT) ☐ 2.2.9 ☐ 2.1.17 (SNAPSHOT) ☐ 2.1.16

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 14 ☒ 11 ☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

No dependency selected

GENERATE CTRL + ⌘

EXPLORE CTRL + SPACE

SHARE...

- 어플리케이션 개발에 필수 요소들만 모아두었다.
- 간단한 설정으로 개발 및 커스텀이 가능하다.
- 간단하고, 빠르게 어플리케이션 실행 및 배포가 가능하다.
- 대규모프로젝트(운영환경)에 필요한 비 기능적 기능도 제공한다.
- 오랜 경험에서 나오는 안정적인 운영이 가능하다.
- Spring에서 불편한 설정이 없어졌다. (XML 설정 등등)

Chapter 04. Spring Boot

# REST Client 설치 하기



[https://chrome.google.com/webstore/category/extensions?utm\\_source=chrome-ntp-icon](https://chrome.google.com/webstore/category/extensions?utm_source=chrome-ntp-icon)

[https://chrome.google.com/webstore/detail/talend-api-tester-free-ed/aejoelaoggembcahagimdiliamlcdmfm?utm\\_source=chrome-ntp-icon](https://chrome.google.com/webstore/detail/talend-api-tester-free-ed/aejoelaoggembcahagimdiliamlcdmfm?utm_source=chrome-ntp-icon)

## Chapter 04. Spring Boot

# Hello Spring Boot

## Chapter 04. Spring Boot

# GET API

	의미	CRUD	역등성	안정성	Path Variable	Query Parameter	DataBody
GET	리소스 취득	R	O	O	O	O	X
POST	리소스 생성, 추가	C	X	X	O	△	O
PUT	리소스 갱신, 생성	C / U	O	X	O	△	O
DELETE	리소스 삭제	D	O	X	O	O	X
HEAD	헤더 데이터 취득	-	O	O	-	-	-
OPTIONS	지원하는 메소드 취득	-	O	-	-	-	-
TRACE	요청메시지 반환	-	O	-	-	-	-
CONNECT	프록시 동작의 터널 접속으로 변경	-	X	-	-	-	-

@GetMapping	Get Resource 설정
@RequestParam	URL Query Param Parsing
@PathVariable	URL Path Variable Parsing
Object	Query Param Object로 Parsing

@RestController	Rest API 설정
@RequestMapping	리소스를 설정 (method로 구분가능)

## Chapter 04. Spring Boot

# POST API

	의미	CRUD	역등성	안정성	Path Variable	Query Parameter	DataBody
GET	리소스 취득	R	O	O	O	O	X
POST	리소스 생성, 추가	C	X	X	O	△	O
PUT	리소스 갱신, 생성	C / U	O	X	O	△	O
DELETE	리소스 삭제	D	O	X	O	O	X
HEAD	헤더 데이터 취득	-	O	O	-	-	-
OPTIONS	지원하는 메소드 취득	-	O	-	-	-	-
TRACE	요청메시지 반환	-	O	-	-	-	-
CONNECT	프록시 동작의 터널 접속으로 변경	-	X	-	-	-	-

@RestController	Rest API 설정
@RequestMapping	리소스를 설정 (method로 구분가능)

@PostMapping	Post Resource 설정
@RequestBody	Request Body 부분 Parsing
@PathVariable	URL Path Variable Parsing
@JsonProperty	json naming
@JsonNaming	class json naming



## Chapter 04. Spring Boot

# PUT API

	의미	CRUD	역등성	안정성	Path Variable	Query Parameter	DataBody
GET	리소스 취득	R	O	O	O	O	X
POST	리소스 생성, 추가	C	X	X	O	△	O
PUT	리소스 갱신, 생성	C / U	O	X	O	△	O
DELETE	리소스 삭제	D	O	X	O	O	X
HEAD	헤더 데이터 취득	-	O	O	-	-	-
OPTIONS	지원하는 메소드 취득	-	O	-	-	-	-
TRACE	요청메시지 반환	-	O	-	-	-	-
CONNECT	프록시 동작의 터널 접속으로 변경	-	X	-	-	-	-

@RestController	Rest API 설정
@RequestMapping	리소스를 설정 (method로 구분가능)

@PutMapping	Put Resource 설정
@RequestBody	Request Body 부분 Parsing
@PathVariable	URL Path Variable Parsing

## Chapter 04. Spring Boot

# DELETE API

	의미	CRUD	역등성	안정성	Path Variable	Query Parameter	DataBody
GET	리소스 취득	R	O	O	O	O	X
POST	리소스 생성, 추가	C	X	X	O	△	O
PUT	리소스 갱신, 생성	C / U	O	X	O	△	O
DELETE	리소스 삭제	D	O	X	O	O	X
HEAD	헤더 데이터 취득	-	O	O	-	-	-
OPTIONS	지원하는 메소드 취득	-	O	-	-	-	-
TRACE	요청메시지 반환	-	O	-	-	-	-
CONNECT	프록시 동작의 터널 접속으로 변경	-	X	-	-	-	-

@RestController	Rest API 설정
@RequestMapping	리소스를 설정 (method로 구분가능)

@DeleteMapping	Delete Resource 설정
@RequestParam	URL Query Param Parsing
@PathVariable	URL Path Variable Parsing
Object	Query Param Object로 Parsing

Chapter 04. Spring Boot

# Response 내려주기

String	일반 Text Type 응답
Object	자동으로 Json 변환되어 응답 상태값은 항상 200 OK
ResponseEntity	Body의 내용을 Object로 설정상황에 따라서 Http Status Code설정
@ResponseBody	RestController가 아닌 곳(Controller)에서Json 응답을 내릴 때



## Chapter 04. Spring Boot

# 모범 사례