

우아한 스프링 부트

백기선 (A.K.A. W H I T E S H I P)

"스프링 부트"가 다루는 영역

애플리케이션 개발 관련 거의 모든 영역

빌드

- 프로젝트 생성
- 의존성 관리
- 애플리케이션 패키징 및 실행

코딩

- 개발 툴 제공
- 자동 설정
- 외부 설정

배포 및 관리

- 도커 이미지 생성
- Actuator
- 스프링 부트 어드민

빌드: 프로젝트 생성

스프링 애플리케이션 개발 시작은? start.spring.io

- 메이븐, 그라들
- 자바, 코틀린, 그루비
- 의존성 추가
- 프로젝트 기본 정보 입력
- 패키징 방법 선택
- 자바 버전 선택

The screenshot shows the Spring Initializr web application interface. The browser tab is titled "Spring Initializr" and the address bar shows "start.spring.io". The interface is dark-themed and contains the following sections:

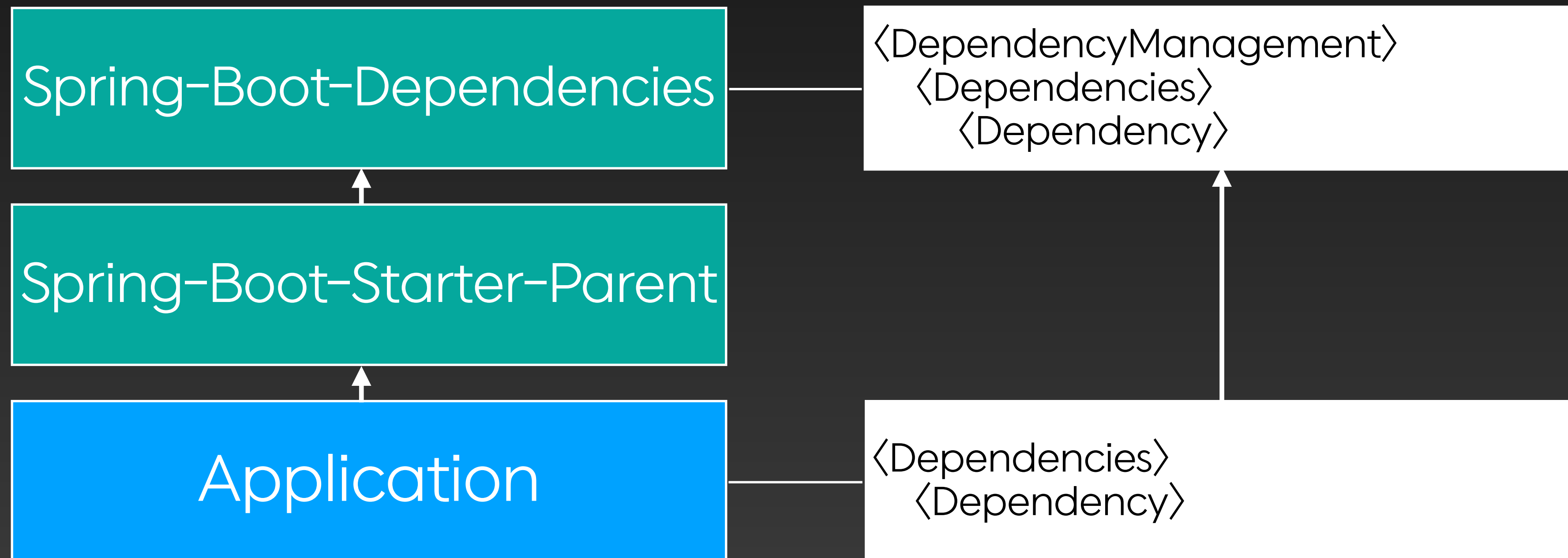
- Project:** Includes radio buttons for "Maven Project" (selected) and "Gradle Project".
- Language:** Includes radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for versions: "2.5.0 (SNAPSHOT)", "2.4.2 (SNAPSHOT)", "2.4.1" (selected), "2.3.8 (SNAPSHOT)", and "2.3.7".
- Project Metadata:** Includes input fields for "Group" (com.example), "Artifact" (demo), "Name" (demo), "Description" (Demo project for Spring Boot), and "Package name" (com.example.demo). It also includes a "Packaging" section with radio buttons for "Jar" (selected) and "War", and a "Java" version section with radio buttons for "15", "11" (selected), and "8".
- Dependencies:** Includes a button "ADD DEPENDENCIES... ⌘ + B". Below it, there are two sections: "Spring Web" with a "WEB" tag and description "Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.", and "Lombok" with a "DEVELOPER TOOLS" tag and description "Java annotation library which helps to reduce boilerplate code.".

At the bottom of the interface, there are three buttons: "GENERATE ⌘ + ↵", "EXPLORE CTRL + SPACE", and "SHARE...".

빌드: 의존성 관리

프로젝트에 필요한 의존성 쉽게 관리하기

- 버전을 적지 않았는데 어떻게 스프링 버전 5.*가 의존성으로 들어와 있을까?
- 메이븐의 <parent>, <DependencyManagement> 그리고 <properties>를 활용한 의존성 관리.



빌드: 애플리케이션 실행

다양한 애플리케이션 실행 방법

mvn spring-boot:run

- 스프링 부트 메이븐 플러그인 사용.
- 메이븐을 사용해서 실행한다.

main 클래스 실행

- 가장 평범한 자바 애플리케이션 실행하는 방법.
- IDE 사용해서 실행한다.

JAR 패키징 & java -jar

- 스프링 부트 플러그인을 사용해 특수한 실행 가능한 형태의 JAR 파일 만들 수 있다.
- 서버에 배포하는 용도로 주로 사용한다.

코딩: 개발 툴

개발 중에 유용한 기능 제공

- 개발 중에 뷰 리소스 또는 템플릿에 적용되는 캐시는 오히려 불편하다.
- 개발 중에 애플리케이션을 자주 재시작 한다.
- 개발 중에 웹 브라우저를 자주 리로딩 한다.

Spring-Boot-Devtools

코딩: 자동 설정

Convention over Configuration

애플리케이션에서 설정한 빈 등록

- @ComponentScan
- @Component, @Service, @Controller, @Repository
- @Configuration
- @Bean

자동 설정으로 제공하는 빈 등록

- META-INF/spring.factories
- EnableAutoConfiguration
- @Configuration && @ConditionalOnXxx

코딩: 외부 설정 파일

코드에서 값을 밖으로 꺼내는 방법 제공

- **application.properties** 또는 **application.yaml**, 환경 변수, java 명령어 아규먼트 등 키/값의 형태로 정의되어 있는 다양한 외부 설정을 지원한다.
- application.properties
 - 가장 구체적이고 가까운 위치에 있는 설정의 우선 순위가 높다.



배포: 도커 이미지 빌드

계층형 이미지 빌드 지원

- 도커 이미지는 다른 이미지를 기반으로 새로운 이미지를 만들 수 있다.
- 계층형 이미지를 만든다면 기존 계층은 캐시로 재사용할 수 있어 효율적이다.



관리: Actuator

애플리케이션 관련 데이터 및 모니터링 정보 제공

Spring-Boot-Starter-Actuator

- 웹(JSON)과 JMX 지원
- 여러 엔드포인트 제공
 - /beans “빈” 정보 조회.
 - /configprops “프로퍼티” 정보 조회.
 - /logger “로거” 정보 조회 및 변경 가능.
 - /heapdump 메모리의 현재 상태를 내려 받을 수 있다.
 - /threaddump 쓰레드의 현재 상태를 내려 받을 수 있다.
 - 이밖에도 /metrics, /mappings 등 여러 엔드포인트를 제공한다.

관리: 스프링 부트 어드민

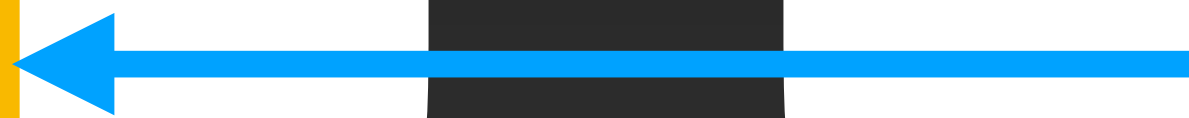
스프링 부트 Actuator 기반 UI

Admin Server

Admin

Application

Client



<https://github.com/codecentric/spring-boot-admin>

감사합니다.