



Spring Boot Basic

스프링 부트 기초

스프링 부트 스타터



한국기술교육대학교
온라인평생교육원

학습내용

- 메이븐을 이용한 라이브러리 의존성 관리
- 스프링 부트 스타터(Starter) 이해
- 사용자 정의 스타터 작성

학습목표

- **메이븐**을 이용하여 스프링 프로젝트의 **라이브러리 의존성을 해결**할 수 있다.
- **스프링 부트 스타터**의 개념을 설명할 수 있다.
- **사용자 정의 스타터를 작성**하고 적용할 수 있다.



메이븐을 이용한 라이브러리 의존성 관리

메이븐을 이용한 라이브러리 의존성 관리

1 메이븐 개념

① 메이븐이란?

- 아파치에서 만든 JAVA 전용의 빌드 자동화 도구로서 Ant의 대안으로 등장
- 애플리케이션 개발과 운용에 필요한 라이브러리 뿐 아니라 라이브러리 간의 의존성 문제도 해결



메이븐을 이용한 라이브러리 의존성 관리

1 메이븐 개념

② POM(Project Object Model)

- pom.xml 파일은 Project Object Model 정보를 담고 있음
- pom.xml에는 프로젝트 정보, 빌드 설정, 빌드 환경, pom 연관 정보 등이 기술됨



메이븐을 이용한 라이브러리 의존성 관리

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

① 스프링 부트와 메이븐

- 스프링 부트 프로젝트를 생성할 때 Type을 Maven으로 설정하면 메이븐 기반의 프로젝트가 생성됨

Type:	Maven	▼	Jar	▼
Java Version:	11	▼	Language:	Java

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

② 메이븐 리포지터리

- pom.xml 파일을 통해 프로젝트에 필요한 라이브러리 의존성을 관리
- 메이븐 리포지터리 사이트(<https://mvnrepository.com>)를 통해 프로젝트에 필요한 라이브러리를 검색

메이븐을 이용한 라이브러리 의존성 관리

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

② 메이븐 리포지터리

Project Lombok » 1.18.20
Spice up your java: Automatic Resource Management, autom

License	MIT
HomePage	https://projectlombok.org
Date	(Apr 02, 2021)
Files	jar (1.8 MB) View All
Repositories	Central ICM
Used By	12,441 artifacts

Maven Gradle SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.20</version>
  <scope>provided</scope>
</dependency>
```

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

③ pom.xml 설정

- pom.xml에 등록된 라이브러리는 물리적으로 사용자 컴퓨터의 Local Repository에 다운로드 되고 'Maven Dependencies'에 등록

```
> logback-classic-1.2.3.jar - C:\Users#gurum#.m2#repository#ch#qos#logback#logback-classic#1.2.3
> logback-core-1.2.3.jar - C:\Users#gurum#.m2#repository#ch#qos#logback#logback-core#1.2.3
> lombok-1.18.20.jar - C:\Users#gurum#.m2#repository#org#projectlombok#lombok#1.18.20
> mockito-core-3.9.0.jar - C:\Users#gurum#.m2#repository#org#mockito#mockito-core#3.9.0
> mockito-junit-jupiter-3.9.0.jar - C:\Users#gurum#.m2#repository#org#mockito#mockito-junit-jupiter#3.9.0
```




메이븐을 이용한 라이브러리 의존성 관리

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

③ pom.xml 설정

- 라이브러리를 추가할 때 Edit Starters...를 이용할 수도 있음

```
<> dependency
# comment - xml comment
# XSL processing instruction - XSL processing instruction
Insert dependency
Add Starters...
```

메이븐을 이용한 라이브러리 의존성 관리

2 스프링 부트 프로젝트와 메이븐

④ POM 상속과 라이브러리 버전 관리

- 부모의 POM 설정을 재정의(Overriding)할 수 있음

```
<properties>
  <java.version>11</java.version>
  <lombok.version>1.18.18</lombok.version>
</properties>
```



스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

① 스타터 개념

- 스프링 부트 프로젝트는 메이븐을 기반으로 프로젝트에서 필요한 의존성을 관리함
- 스프링 부트의 스타터는 필요한 라이브러리들을 묶어서 패키지 형태로 제공함

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

① 스타터 개념

- 일반적으로 스타터는 관련된 라이브러리 의존성이 설정되지만 또 다른 스타터를 포함하기도 함
- 스프링 부트는 다양한 모듈의 스타터를 제공하며 “spring-boot-starter-모듈명” 형태의 이름을 가짐



스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

② 스타터 관리

- 스타터는 스프링 부트 프로젝트를 생성할 때 추가함
- 프로젝트가 생성된 후에도 Edit Starters...를 이용하여 추가할 수 있음

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

② 스타터 관리

- ▶ Alibaba
- ▶ Amazon Web Services
- ▶ Developer Tools
- ▶ Google Cloud Platform
- ▶ I/O
- ▶ Messaging
- ▶ Microsoft Azure
- ▶ NoSQL
- ▶ Observability
- ▶ Ops
- ▶ SQL
- ▶ Security
- ▶ Spring Cloud
- ▶ Spring Cloud Circuit Breaker
- ▶ Spring Cloud Config
- ▶ Spring Cloud Discovery
- ▶ Spring Cloud Messaging

- Lombok이나 DevTools 같은 스타터가 아닌 단순 라이브러리도 있음



스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

③ 기본 스타터

- 스프링 웹(Spring Web)모듈을 선택하고 스프링 부트 프로젝트를 생성하면 pom.xml에 기본적으로 spring-boot-starter-web과 spring-boot-starter-test 스타터가 추가됨

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

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

스프링 부트 스타터 (Starter) 이해

1 스프링 부트 스타터 (Starter) 이해

③ 기본 스타터

- spring-boot-starter-web은 Spring MVC 관련 모듈을 포함하여 웹 애플리케이션 개발에 필요한 라이브러리들을 제공함
- spring-boot-starter-test는 JUnit을 기반으로 하는 테스트 케이스와 테스트 스위트를 위한 라이브러리들을 제공함



스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

2 스프링 기반의 웹 애플리케이션

① 스프링 기반의 웹 애플리케이션 구조 이해

spring-boot-starter-web 스타터

- 세 개의 스타터
- 두 개의 라이브러리에 대한 의존성 설정

스프링 부트 스타터 (Starter) 이해

2 스프링 기반의 웹 애플리케이션

① 스프링 기반의 웹 애플리케이션 구조 이해

spring-boot-starter-web 스타터

```
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-json</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
</dependencies>
```

스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

2 스프링 기반의 웹 애플리케이션

① 스프링 기반의 웹 애플리케이션

spring-boot-starter

```
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-json</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
</dependencies>
```

```
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-json</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<version>2.5.2</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.8</version>
<scope>compile</scope>
</dependency>
</dependencies>
```

스프링 부트 스타터 (Starter) 이해

2 스프링 기반의 웹 애플리케이션

② 스타터 재정의

- 스타터는 pom.xml 파일에서 재정의할 수 있음
- 특정 라이브러리 버전만 변경하기 위해서는 해당 라이브러리의 <version> 설정을 재정의

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.7</version>
</dependency>
```

스프링 부트 스타터 (Starter) 이해

스프링 부트 스타터 (Starter) 이해

2 스프링 기반의 웹 애플리케이션

② 스타터 재정의

- Java 버전이나 Spring의 모든 라이브러리 버전을 변경하기 위해서는 <properties> 설정을 재정의

```
<properties>
  <java.version>11</java.version>
  <spring-framework.version>5.3.7</spring-framework.version>
</properties>
```




사용자 정의 스타터 작성

사용자 정의 스타터 작성

1 사용자 정의 스타터

① 사용자 정의 스타터란?

- 스프링 프레임워크는 범용적인 프로젝트 개발을 목표로 설계되었기 때문에 특정 도메인이나 비즈니스에 최적화되어 있지 않음

사용자 정의 스타터 작성

1 사용자 정의 스타터

① 사용자 정의 스타터란?

- 스프링 부트가 제공하는 스타터가 아닌 사용자가 스타터를 정의하여 사용할 수 있음
- 일반적으로 사용자 정의 스타터의 이름은 XXX-spring-boot-starter 같은 패턴을 따름

사용자 정의 스타터 작성

사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

- 사용자 정의 스타터는 메이븐 프로젝트로 생성

사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

1 메이븐 프로젝트 생성(board-spring-boot-starter)

2 pom.xml 파일에 <dependency> 등록

3 메이븐 로컬 저장소에 사용자 정의 스타터 등록

4 프로젝트에서 사용자 정의 스타터 추가



사용자 정의 스타터 작성

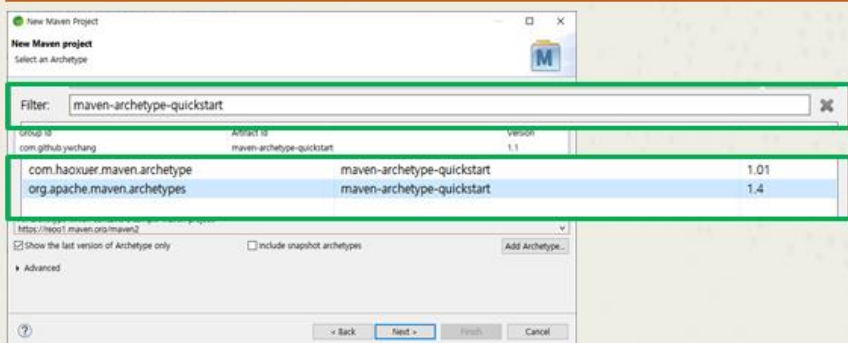
사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

1 메이븐 프로젝트 생성

- Filter에 maven-archetype-quickstart를 입력하고 메이븐에서 제공하는 기본 템플릿을 선택



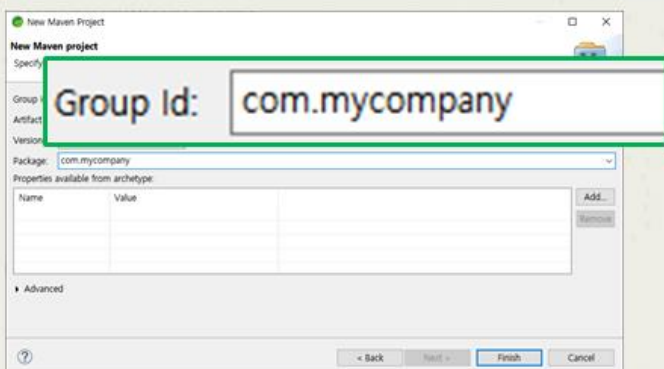
사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

1 메이븐 프로젝트 생성

- Group Id를 com.mycompany로 지정





사용자 정의 스타터 작성

사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

2 pom.xml 파일에 <dependency> 등록

```

<groupId>com.mycompany</groupId>
<artifactId>board-spring-boot-starter</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>board-spring-boot-starter</name>
<url>http://www.example.com</url>

<dependencies>
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.7</version>
  </dependency>
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.6</version>
  </dependency>
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.4.200</version>
  </dependency>
</dependencies>

```

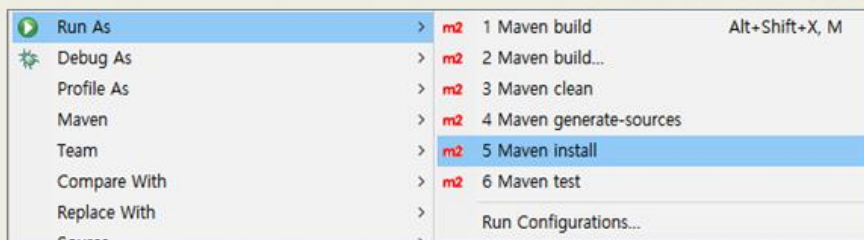
사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

3 메이븐 로컬 저장소에 사용자 정의 스타터 등록

• Run As → Maven install 명령을 실행





사용자 정의 스타터 작성

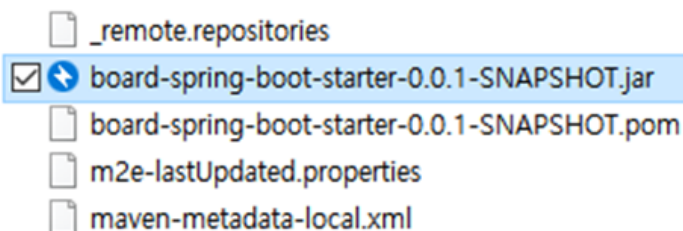
사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

3 메이븐 로컬 저장소에 사용자 정의 스타터 등록

- 물리적으로 board-spring-boot-starter-0.0.1-SNAPSHOT.jar 파일이 등록됨



사용자 정의 스타터 작성

1 사용자 정의 스타터

② 사용자 정의 스타터 작성

4 사용자 정의 스타터 적용

- 작업중인 프로젝트 pom.xml 파일에 사용자 정의 스타터의 <dependency>를 추가

```
<dependency>
  <groupId>com.mycompany</groupId>
  <artifactId>board-spring-boot-starter</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>
</dependencies>
```




실전 스프링 부트

스프링 부트 스타터 // 실전 스프링 부트

Spring Boot Basic

```

1 package com.mycompany;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.WebApplicationType;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6
7 @SpringBootApplication
8 public class HelloBootApplication {
9
10     public static void main(String[] args) {
11         SpringApplication.run(HelloBootApplication.class, args);
12         SpringApplication application = new SpringApplication(HelloBootApplication.class);
13         application.setWebApplicationType(WebApplicationType.NONE);
14         application.run(args);
15     }
16
17 }
18

```

1. Maven을 이용한 라이브러리 의존성 관리

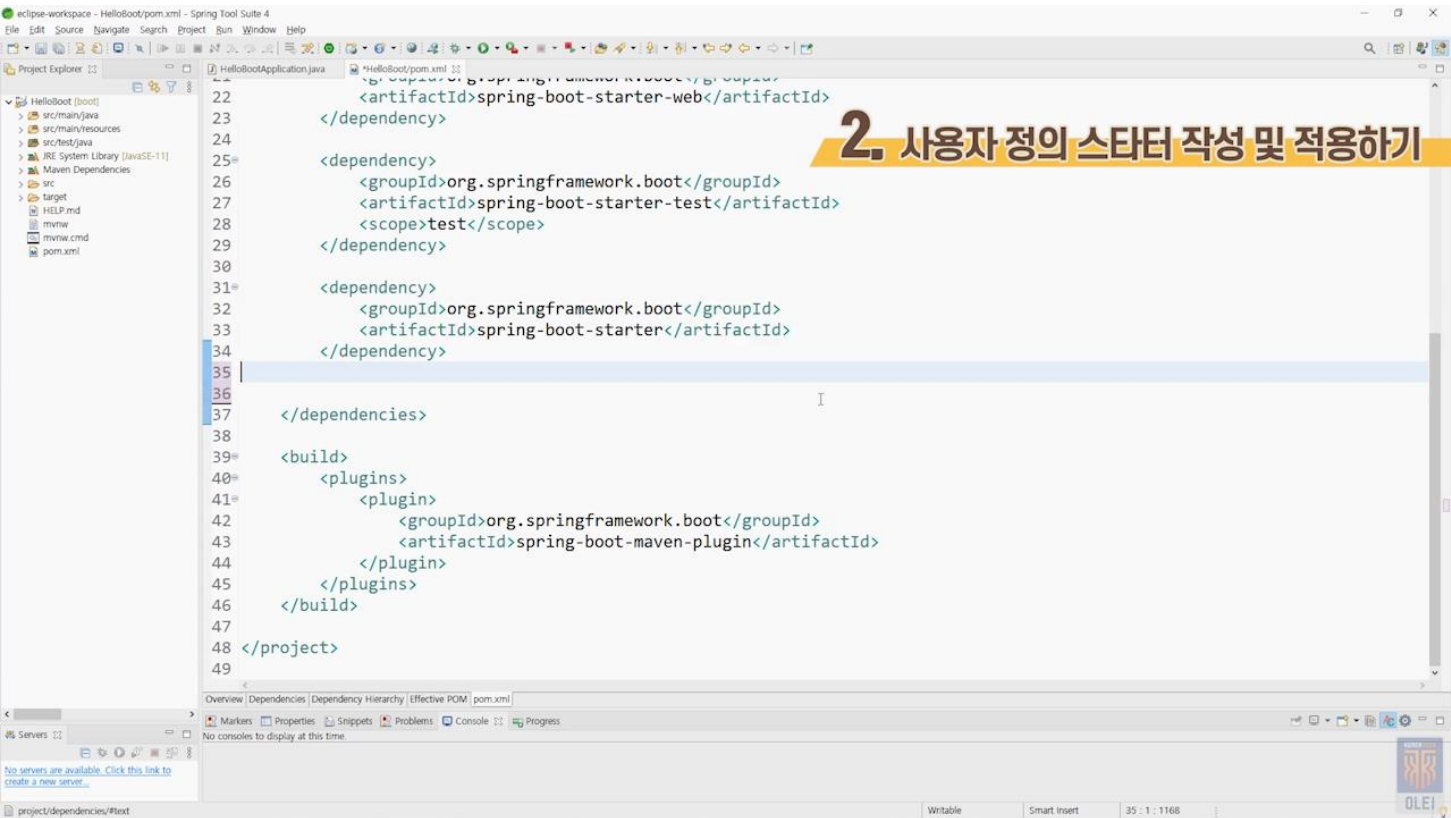
- Zulu(OpenJDK) 16 사용
- Spring Tools 4 for Eclipse 4.0.10 사용

메이븐을 이용한 라이브러리 관리

실습단계
메이븐을 이용한 라이브러리 관리
스프링 부트 제공 스타터의 이해와 사용자 정의 스타터 생성, 라이브러리 관리 처리
pom.xml : 메이븐 환경설정 파일
dependency 설정 하나 당 라이브러리 하나 다운로드
H2 Database 제공 드라이버 다운로드 방법 2가지
1. mvnrepository.com에서 H2 키워드로 검색 → H2 Database Engine
주석 해지 → 실제 dependency만 복사 → pom.xml 파일에 추가
매번 사이트에 들어가서 필요한 소스를 가져오기 귀찮다는 단점이 있음
2.
</dependencies> 태그 바로 위에, 마지막 </dependency> 아래에 <Ctrl> + <Space> → Add Starters 더블클릭
SQL → H2 Database
pom.xml
H2에 대한 dependency 추가된 것 확인 가능



실전 스프링 부트



2. 사용자 정의 스타터 작성 및 적용하기

실습단계

스타터 : 관련된 라이브러리의 묶음

테스트 모듈에 해당하는 스타터를 제거할 경우, 스타터로 다운로드 한 수많은 라이브러리가 같이 사라지게 됨

웹 관련 스타터를 제거하면 스타터로 다운로드한 웹 관련 라이브러리가 사라짐

스타터는 구현하려는 라이브러리를 패키지 형태로 쉽게 관리하기 위해 제공되는 것

사용자 정의 스타터를 만들기 위해 New → Project 선택

Maven → Maven Project 생성

프로젝트 템플릿이 많기 때문에 maven-archetype-quickstart 검색하여 이용

Group Id 설정, 프로젝트 이름 설정

디폴트로 제공되는 루트 패키지를 프로젝트 이름을 제거한 상태로 만드는 것이 좋음

내가 원하는 라이브러리를 [mvnrepository.com] 사이트에서 검색

Mybatis, Mybatis Spring, H2 Database 추가

들여쓰기 : <Ctrl> + <Shift> + <F>

에러가 난 이유 : 버전을 명시하지 않았기 때문, 버전을 지정해야 함



실전 스프링 부트

실습단계
Maven Dependencies에 직접 추가한 라이브러리를 확인할 수 있음
세 가지 정보 복사
board-spring-boot-starter → Run As → Maven install
HelloBoot → pom.xml
마지막 </dependency> 아래에 <dependency> 추가 후 복사한 내용 붙여넣기
사용자 정의 스타터가 추가된 것을 확인할 수 있음 (board-spring-boot-starter-0.0.1-SNAPSHOT.jar)