Spring Boot

eygou@induk.ac.kr lms.induk.ac.kr

SpringBoot

■등장 배경

• Spring 개발 과정에서 빈번하게 발생하는 개발자의 결정 요구로 개발 부담이 증대되어 이에 대한 개선 방안이 요구됨

-목표

- 모든 Spring 개발을 위한 급격하게 빠르고, 광범위하게 접근할 수 있는 시작 경험을 제공한다.
- 설치 과정 없이 바로 사용하도록 의견을 제시하지만 요 구들이 기본값들로부터 분산되기 시작하면 신속하게 벗 어나야 한다.
- 대규모 프로젝트 클래스를 위한 공통적인 다양한 비 기 능적 특징(임베디드 서버, 보안, 메트릭, 상태 확인 및 외 부화된 구성)을 제공한다.
- 절대로 XML 구성에 대한 코드 생성이 없고 요구 사항이 없다.

■정의

- Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".
- 독립실행형(stand-alone) 프로덕션급(production-grade)
 Spring 기반 응용 프로그램 개발을 신속하고 쉽게 만들기 위한
 Spring의 CoC(convention-over-configuration) 솔루션
 - 최소한의 설정으로 Spring 플랫폼과 서드파티(third-party) 라이브 러리들을 사용할 수 있도록 함
 - 유연성을 잃지 않고 프레임워크를 사용하는 개발자가 결정해야하는 의사 결정 횟수를 줄이려는 시도

■특징

- 독립실행가능한 스프링 애플리케이션을 생성한다.
- 내장형 Tomcat, Jetty 또는 Undertow를 포함하고 있어 WAR 파일을 배포가 필요하지 않는다.
- 빌드 구성(configuration)을 간소화하기 위한 독자적인 'starter dependencies'를 제공한다.
- 가능한 경우마다 자동으로 Spring과 서드파티 라이브러리 구성을 수 행한다.
- 메트릭, 상태 확인 및 외부 구성과 같은 프로덕션 준비 기능을 제공합니다.
- 코드 생성 및 XML 구성 요구 사항이 없다.
- 참고) https://spring.io/projects/spring-boot



Why Spring ~

Learn ∨

Projects ~

Training

Support

Community ~



Spring Boot 2.6.4







OVERVIEW

LEARN

SUPPORT

SAMPLES

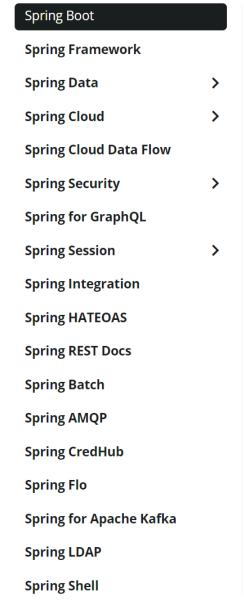
Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out the project release notes section on our wiki.

Features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration



- SpringBoot 2.6.x GA (2021/11/19)
 - Spring 5.3.x 지원, JDK 16 지원
 - Java 8+, Maven 3.3 +, Gradle 7.0.x+
 - 임베디드 서블릿 컨테이너
 - Tomcat 9.0(4.0), Jetty 10.x(Java 11이상 필요), Undertow 2.0(4.0)
 - 참고: https://github.com/spring-projects/springboot/wiki#release-notes

- Spring project 변화
 - Spring Security 5.6
 - Spring Data 2021.1
 - Spring HATEOAS 1.4
 - Spring Kafka 2.8
 - Spring AMQP 2.4
 - Spring Session 2021.1.0

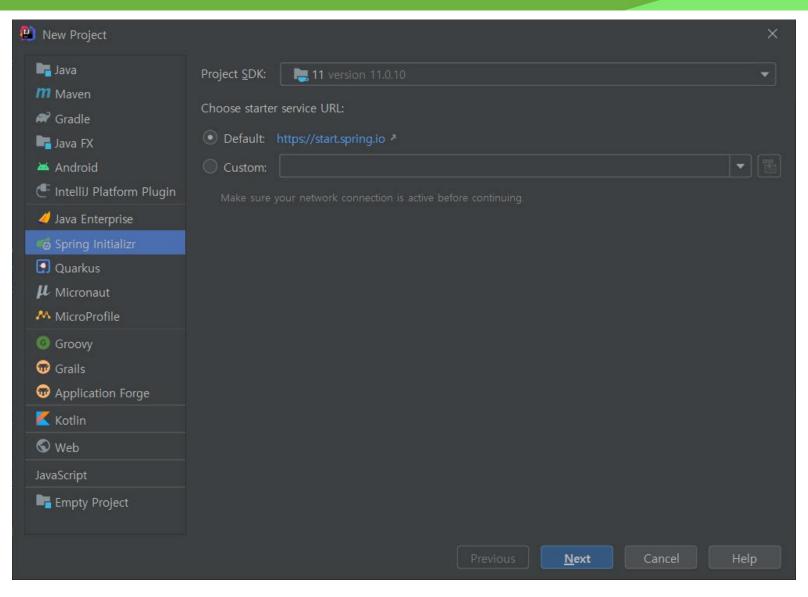
계속 - 기본 pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.
   <modelVersion>4.0.0</modelVersion>
   <groupId>com.example
   <artifactId>myproject</artifactId>
   <version>0.0.1-SNAPSHOT</version>
   <!-- Inherit defaults from Spring Boot -->
   <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>2.2.5.RELEASE
   </parent>
```

```
<!-- Override inherited settings -->
<description/>
<developers>
    <developer/>
</developers>
clicenses>
   <license/>
</licenses>
<scm>
   <url/>
</scm>
<url/>
<!-- Add typical dependencies for a web application -->
<dependencies>
    <dependency>
       <groupId>org.springframework.boot
       <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
```

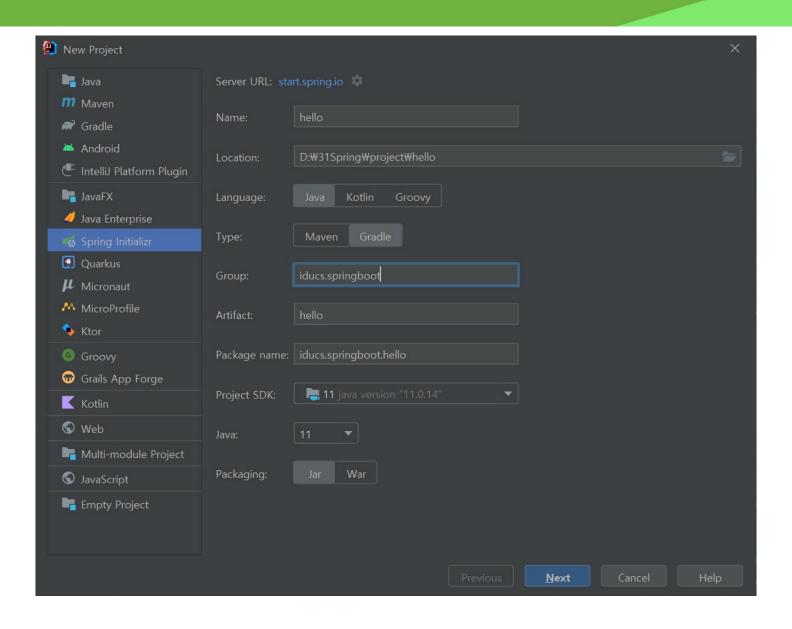
```
<!-- Package as an executable jar -->
   <build>
       <plugins>
           <plugin>
               <groupId>org.springframework.boot
               <artifactId>spring-boot-maven-plugin</artifactId>
           </plugin>
       </plugins>
   </build>
</project>
```

New Spring Starter Project



■항목 설명

- Service URL
 - 스프링 부트 프로젝트의 빠른 시작을 위한 방법을 제공하는 확장 API의 주소를 지정
 - 설정 값 https://start.spring.io/
- Name
 - 스프링 부트 프로젝트 이름 지정
 - ■복합 단어가 나올 때 '-'기호를 사용한다.



- Group
 - 모든 프로젝트들 사이에게 고유하게 식별하는데 사용할 그룹 아이디를 지정
 - package 명명 규칙을 준수하여 함
 - 형식: <도메인 네임>.<그룹명>[.<그룹명>]
- Artifact
 - 산출물 이름, 버전정보를 생략한 jar 파일의 이름

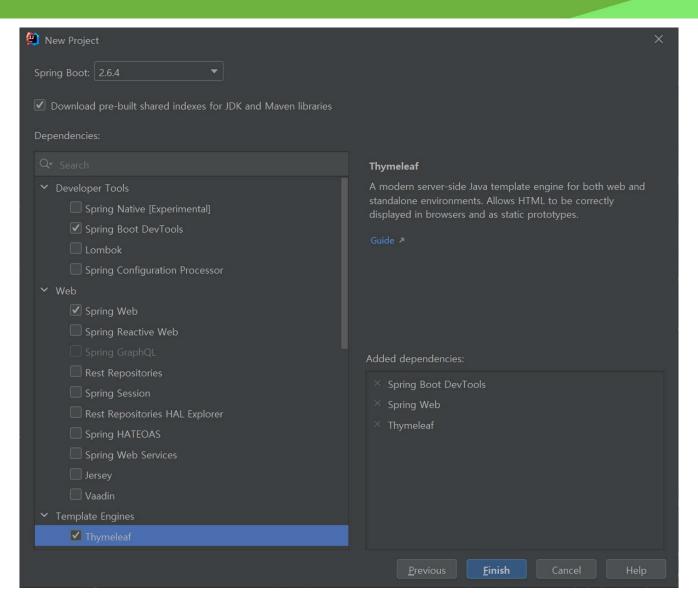
- Type: Build Model 유형 선택
 - Maven

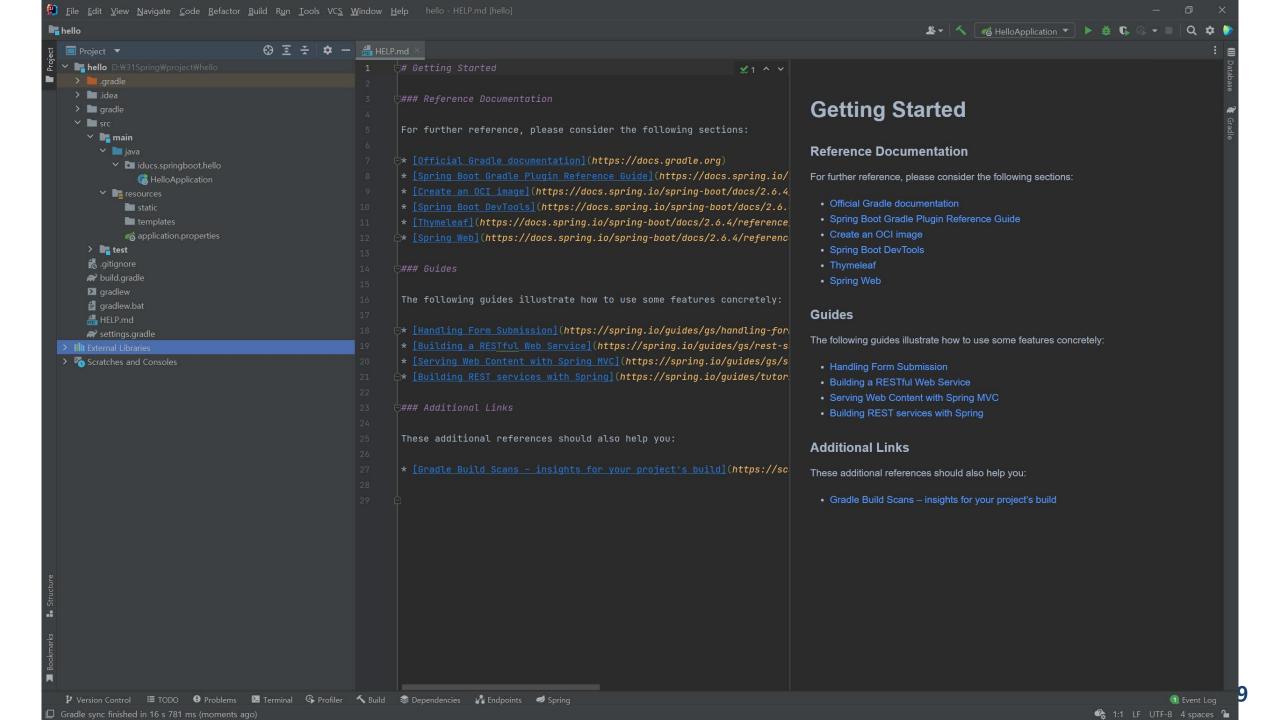
```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

- Gradle
 - 'org.springframework.boot:spring-boot-starter-data-jpa'
- Packaging : 패키지 형식 선택
 - Jar: 독립실행형
 - War : 웹 배포형

- Version : 산출물 버전
- Name : 산출물 이름
- Description : 산출물에 대한 설명
- Package
 - 프로젝트의 기본 패키지 정보를 지정
 - <groupid>.<artifactid> 나 <groupid>를 사용함

계속 - 의존성 해결

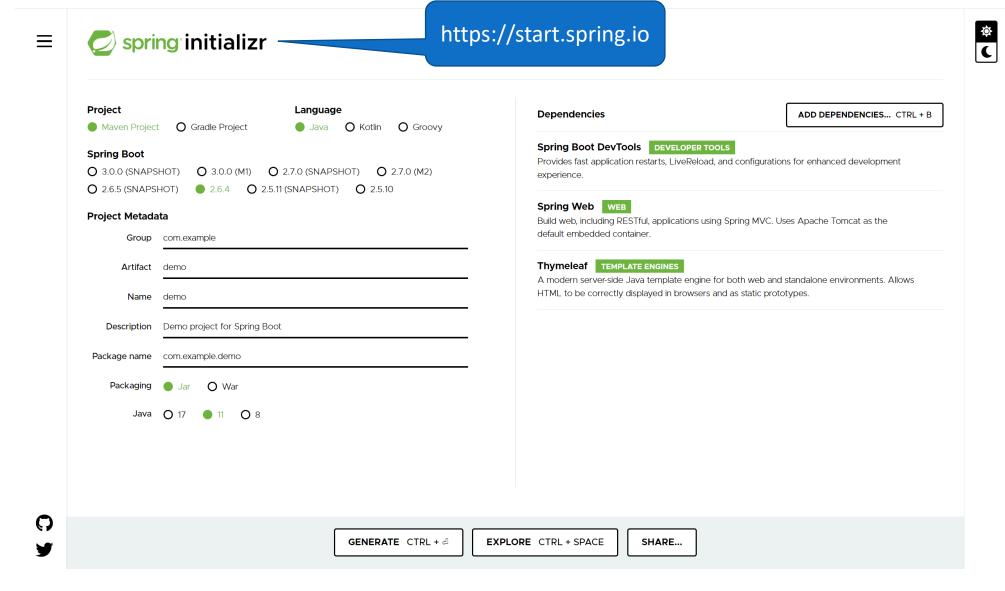




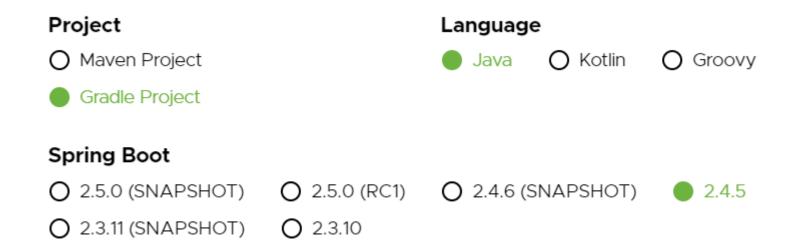
HelloApplication.java

package iducs.springboot.hello; import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RestController; @SpringBootApplication @RestController public class HelloApplication { public static void main(String[] args) { SpringApplication.run(HelloApplication.class, args); @GetMapping("/hello") public String helll() { return "Hello SpringBoot World";

Spring Initializr을 활용한 Project 생성



게속 - Project, Language, Spring Boot



계속 - Project Metadata

Project Metadata	
Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	O Jar • War
Java	O 16

계속 - Dependancies

Dependencies

ADD DEPENDENCIES... CTRL + B

Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Boot DevTools DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

계속 - Dependancies

Web, Security, JPA, Actuator, Devtools... Press Ctrl for multiple adds **DEVELOPER TOOLS** Spring Native [Experimental] Incubating support for compiling Spring applications to native executables using the GraalVM native-image compiler. Spring Boot DevTools Provides fast application restarts, LiveReload, and configurations for enhanced development experience. Lombok Java annotation library which helps to reduce boilerplate code. **Spring Configuration Processor** Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files). WEB Spring Web Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. Spring Reactive Web Build reactive web applications with Spring WebFlux and Netty.

- GENERATE
 - 프로젝트 파일 생성
- EXPLORE
 - 프로젝트 파일 내용 탐색

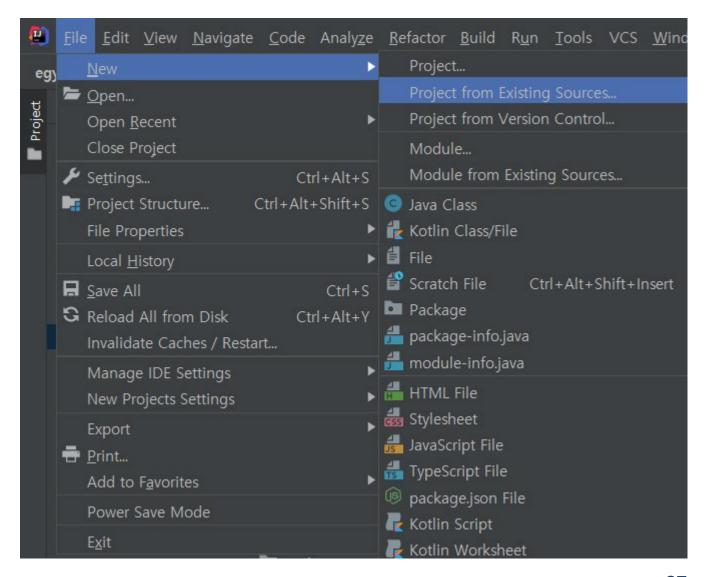
GENERATE CTRL + ₽

EXPLORE CTRL + SPACE

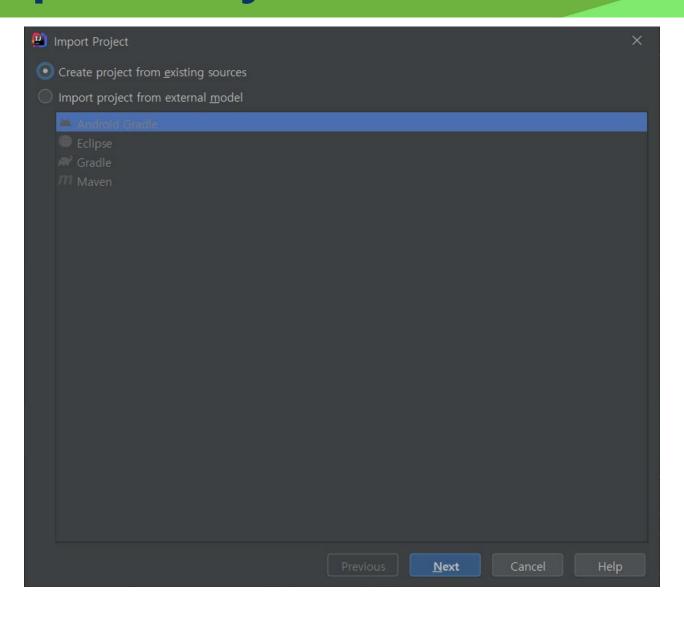
SHARE...

■사용 방법 1

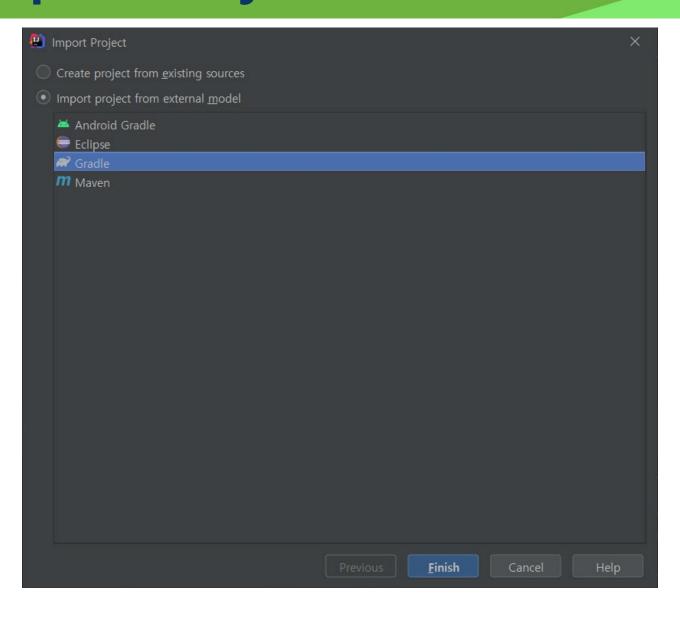
- 생성된 프로젝트를 다운로드 받고, 압 축을 해제
- File > New >
 Project from
 Existing Source 선
 택
- 압축해제한 경로 선택



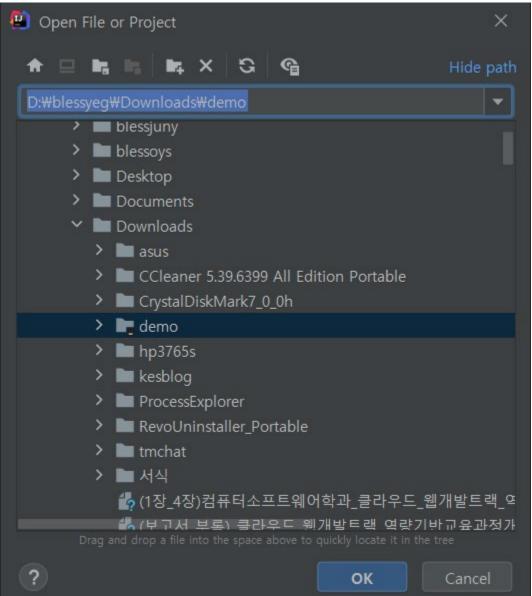
계속 - Import Project (소스로 부터 생성)



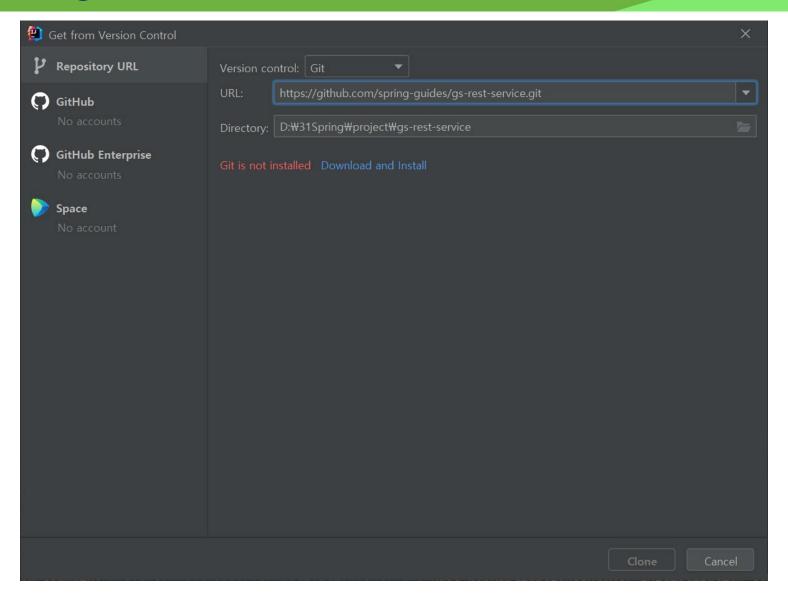
계속 - Import Project (외부 모델로부터 생성)



- ■사용 방법 2
 - File > Open
 - 압축 해제한 프로젝트 선택



new project from version



Spring Boot Project

- ■Spring Stater Project(Spring Boot)의 장점
 - 1.별도의 설정이 필요 없다.
 - 2.WAS없이 실행이 가능하다.
 - 3.로딩 시간이 짧아서 테스트 하기 편하다.

- ■Spring Stater Project(Spring Boot)의 단점
 - 1.기존의 설정과 다른 방식으로 사용한다.
 - 2.JSP 설정 등은 별도로 해야 한다.

Spring Boot 디렉토리 구조

핵심코드 src/main/java iducs/springboot/board ▶ ☐ controller @Controller, @RequestMapping ▶ ☐ domain @Entity @JpaRepository, @CrudRepository ▶ ☐ repository **@Service** src/main/resources js, css, 이미지 파일 ▶ ☐ static 웹 템플릿 코드 **▶ □** templates 테스트 코드 src/test/java

Spring Legacy Project

■Spring Legacy Project의 장점

- 1.현재까지 실무에서 많이 사용되고 있다.
- 2.다양한 자료가 존재한다.
- 3.기존 프로젝트를 이해하는데 도움이 된다.
- 4.모든 버전의 스프링에서 사용할 수 있다.

Spring Legacy Project의 단점

- 1.초반 테스트 환경 구성 등이 어렵다.
- 2.WAS와 연동하는 경우 결과 확인에 많은 리소스를 소모한다.

Spring Legacy 프로젝트의 폴더 구조

- ▷ ☐ src/main/java
 - ▷ iducs/spring/board
 - ▷ ☐ config
 - ▶ ☐ controller
 - ▶ domain
 - ▶ ☐ repository
 - ▷ □ service
- ▷ ☐ src/main/resources
- ▷ ☐ src/main/web
 - ▶ ☐ resources
 - - ▷ □ spring
 - ▶ □ views

- 패키지 정보
- @구성
- @Controller, @RequestMapping
- @Entity
- @JpaRepository, @CrudRepository
- **@Service**

- 정적 자원 : css, js, 이미지 파일
- spring 설정 파일
- 동적 자원: 웹 템플릿 코드

학습 후 기대효과

- ■SpringBoot의 등장 배경, 목표, 정의 및 특징을 설명할 수 있다.
- ■SpringBoot 프로젝트를 생성할 수 있다.
 - Spring Starter Project 활용
 - SpringInitialzr 활용
- ■SpringBoot 프로젝트의 장단점, 디렉터리 구조를 설명할 수 있다.
 - Spring 프로젝트와 비교할 수 있다.