

# SpringBoot 환경세팅

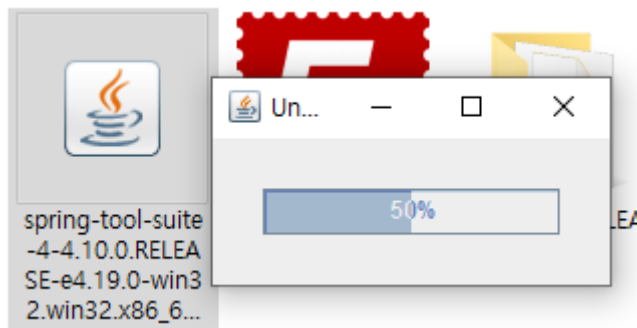
- IDE 프로그램 : 프로그램을 작성할 수 있는 프로그램
- IDE 프로그램으로 이클립스 를 사용해도 되고, STS (Spring Tool Suite)를 사용해도 된다.
  - 이클립스: 스프링 프레임워크를 사용하기 위한 추가적인 플러그인을 따로 설치해야 한다.

## 1. 프로그램 다운로드

### 1) STS(Spring Tool Suite) 다운로드

#### [STS 다운로드 링크](#)

- jar 파일 다운로드 후 아래와 같은 아이콘을 더블클릭한다.
  - 만약 컴퓨터에 알집 프로그램이 설치된 경우 jar 파일이 압축 모양으로 나올 수 있다. 알집을 제거 후 수행한다.



- jar 파일 수행 후 압축이 풀린 sts 폴더가 생성된다.
- 폴더를 원하는 위치에 옮겨둔다.
- SpringToolSuite4.exe 프로그램을 수행하면 된다.

configuration	2021-04-09 오전 11:48	파일 폴더	
dropins	2021-04-09 오전 11:48	파일 폴더	
features	2021-04-09 오전 11:48	파일 폴더	
META-INF	2021-04-09 오전 11:48	파일 폴더	
p2	2021-04-09 오전 11:48	파일 폴더	
plugins	2021-04-09 오전 11:48	파일 폴더	
readme	2021-04-09 오전 11:48	파일 폴더	
.eclipseproduct	2021-04-09 오전 11:48	ECLIPSEPRODUCT...	1KB
artifacts.xml	2021-04-09 오전 11:48	XML 문서	157KB
eclipse.exe	2021-04-09 오전 11:48	응용 프로그램	129KB
license.txt	2021-04-09 오전 11:48	텍스트 문서	12KB
open-source-licenses.txt	2021-04-09 오전 11:48	텍스트 문서	841KB
SpringToolSuite4.exe	2021-04-09 오전 11:48	응용 프로그램	415KB
SpringToolSuite4.ini	2021-04-09 오전 11:48	구성 설정	1KB

## 2. STS 열고 테스트 해보기

### 1) 플러그인 설치

새로운 STS에서는 jsp를 지원하지 않으므로 이클립스 플러그인이 필요하다.

- Help > Eclipse Marketplace... 창에서 **Java and Web** 검색 후 install



## 2) 글자 인코딩 설정

설정하지 않으면 프로그램에서 한글이 깨질 수 있다.

- Window > Preferences 설정창을 띄운다.
  - General > Workspace 에서 좌측 하단 Text file encoding 을 Other 선택 > UTF-8로 변경
  - Web > CSS Files 에서 인코딩 ISO 10646/Unicode(UTF-8) 선택
  - Web > HTML Files 에서 인코딩 ISO 10646/Unicode(UTF-8) 선택
  - Web > JSP Files 에서 인코딩 ISO 10646/Unicode(UTF-8) 선택

## 3) 프로젝트 만들기

- 좌측 Create New Spring Starter Project 또는 File > New > Spring Starter Project 선택

## (1) 프로젝트 기본 설정

**New Spring Starter Project**

Service URL:

Name:

☒ Use default location

Location:

Type:  Packaging:

Java Version:  Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

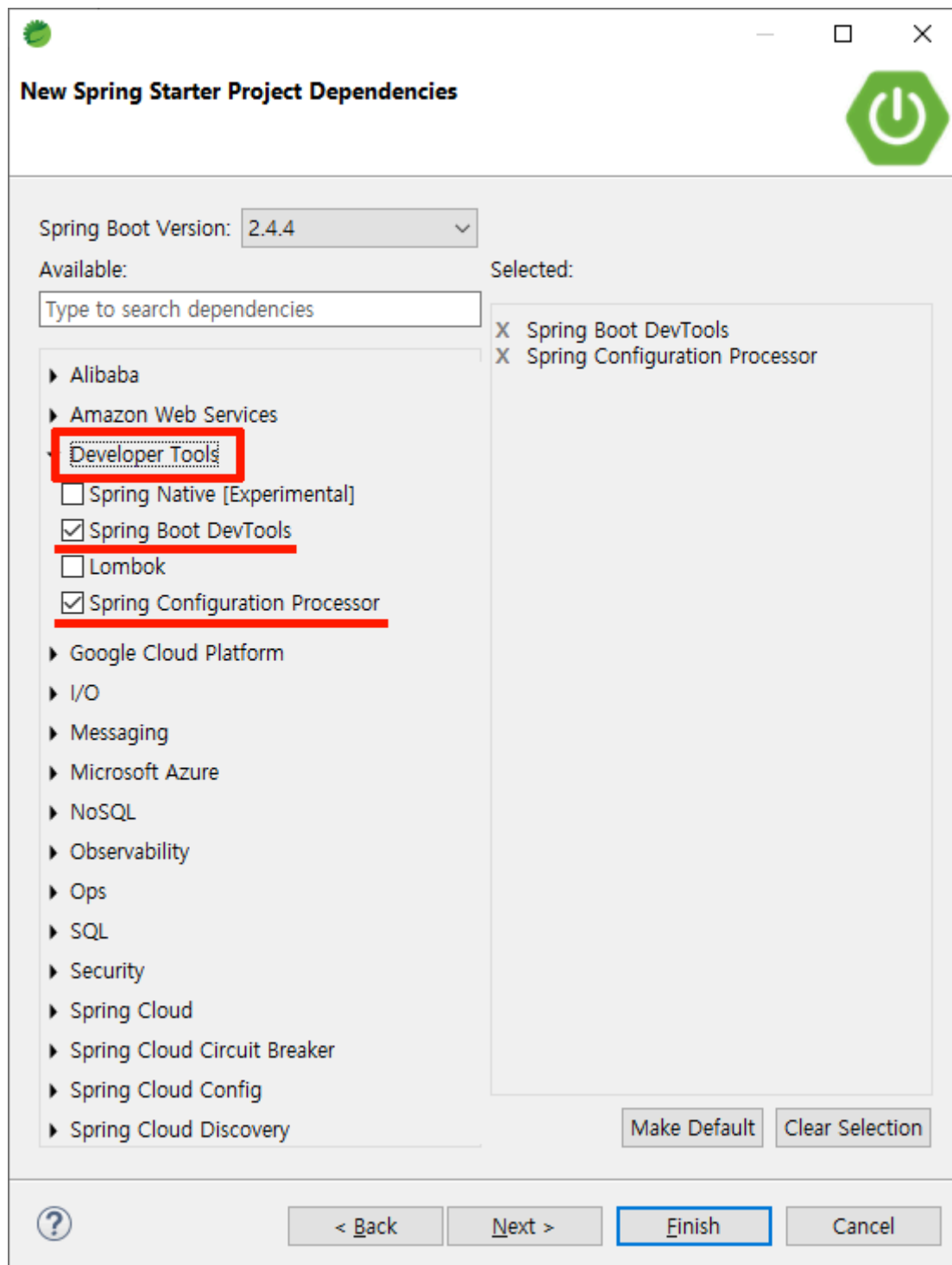
☐ Add project to working sets

Working sets:

- Name: 프로젝트 이름을 정한다.
  - 앞으로 예제와 퀴즈를 다른 프로젝트에서 진행할 예정
  - SpringExample 또는 SpringQuiz
- Type: 빌드 방식 Gradle 선택
  - gradle이 maven보다 속도가 10~100배 빠르고 gradle로 변경되는 추세.
- Group: 만들 사이트의 URL의 도메인 주소(예: <http://marondal.com>) 설정하는 곳
  - 관례적으로 도메인으로 만든다.
  - com. 뒤에 자주 사용하는 id를 넣으면 된다.
- Package: 도메인 거꾸로. 기본 패키지명 형식으로 짓는다.

## (2) 필요한 Dependencies 설정

필요한 라이브러리들을 프로젝트에 연결시킨다.(build.gradle 파일에 라이브러리가 연결됨)



- Spring Boot DevTools : 서버를 켜다꺼지 않아도 변경된 내용을 브라우저에서 즉시 볼 수 있다.
- Spring Configuration Processor : 설정 파일(properties나 yml)에서 자동완성 되도록 도와준다.
- Lombok : 데이터를 저장하기 위한 모델 객체의 Getter, Setter를 어노테이션으로 쉽게 만들 수 있다.

**New Spring Starter Project Dependencies**

Spring Boot Version: 2.4.4

Available:

Type to search dependencies

- ▶ Observability
- ▶ Ops
- SQL**
  - ☐ JDBC API
  - ☐ Spring Data JPA
  - ☐ Spring Data JDBC
  - ☐ Spring Data R2DBC
  - ☒ MyBatis Framework
  - ☐ Liquibase Migration
  - ☐ Flyway Migration
  - ☐ JOOQ Access Layer
  - ☐ IBM DB2 Driver
  - ☐ Apache Derby Database
  - ☐ H2 Database
  - ☐ HyperSQL Database
  - ☐ MariaDB Driver
  - ☐ MS SQL Server Driver
  - ☒ MySQL Driver
  - ☐ Oracle Driver
  - ☐ PostgreSQL Driver
- ▶ Security

Selected:

- X Spring Boot DevTools
- X Spring Configuration Processor
- X MyBatis Framework
- X MySQL Driver

Make Default Clear Selection

? < Back Next > Finish Cancel

- MyBatis Framework : DB 데이터를 xml 형식에 쿼리를 만들어 쉽게 접근할 수 있도록 도와주는 프레임워크
- MySQL Driver : MySQL DB와 프로젝트의 연동을 돕는 드라이버

**New Spring Starter Project Dependencies**

Spring Boot Version: 2.4.4

Available:


Type to search dependencies

- ▶ Spring Cloud Circuit Breaker
- ▶ Spring Cloud Config
- ▶ Spring Cloud Discovery
- ▶ Spring Cloud Messaging
- ▶ Spring Cloud Routing
- ▶ Spring Cloud Security
- ▶ Spring Cloud Tools
- ▶ Template Engines
- ▶ Testing
- ▶ VMware Tanzu Application Service
- Web**
  - ☒ Spring Web
  - ☐ Spring Reactive Web
  - ☐ Rest Repositories
  - ☐ Spring Session
  - ☐ Rest Repositories HAL Explorer
  - ☐ Spring HATEOAS
  - ☐ Spring Web Services
  - ☐ Jersey
  - ☐ Vaadin

Selected:

- X Spring Boot DevTools
- X Spring Configuration Processor
- X MyBatis Framework
- X MySQL Driver
- X Spring Web

Make Default Clear Selection



< Back

Next >

Finish

Cancel

- Spring Web: Tomcat을 컨테이너로 세팅해주고 Spring MVC 구조로 사용할 수 있도록 해준다.

### (3) 코드 동작 시키기

- com.hagulu.ex.hello package를 만든다.
- TestController 클래스를 만든다.
- 아래 코드를 추가하고 서버를 켜다.
- <http://localhost:8080/hello> 로 접속한다.

```
@Controller
public class HelloController {
    @ResponseBody
    @RequestMapping("/hello")
    public String helloWorld() {
        return "Hello World!!";
    }
}
```

## 에러 잡기

### (1) debug 에러

To display the conditions report re-run your application with 'debug' enabled.

Run > Run configurations > Spring boot App > 내 프로젝트 에서 arguments 탭 Program argument 영역에 **--debug** 파라미터 추가

### (2) DB 설정 에러

Failed to determine a suitable driver class

SpringExampleApplication.java에서 클래스명 위에 어노테이션 추가(디비 연동전에 임시로 넣는 코드)

```
@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
```

## 3. JSP View 연동하기

### dependency 추가

build.gradle 에 디펜던시를 추가한다.

필요한 라이브러리들을 검색해서 dependencies에 코드를 추가하고,  
**build.gradle** 우측 마우스 클릭 > **Gradle** > **Gradle Refresh Project** 를 누르면 적용된다.

반드시 **build refresh** 할 것!!! (안하면 404의 늪에 빠진다.)

```
implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'
implementation group: 'org.glassfish.web', name: 'jakarta.servlet.jsp.jstl', version: '2.0.0'
```

### application.properties 추가

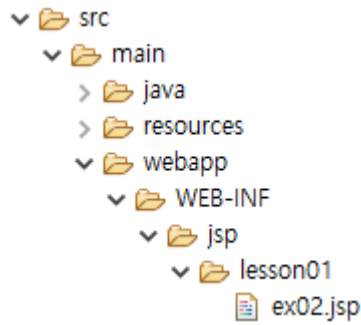
```
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp
```

- server.port: 포트 변경

### jsp 추가

- 톰캣 기반 웹 어플리케이션은 WEB-INF 아래 위치시키면 URL로 직접 접근을 막는다.(보안!)

```
src/main/webapp/WEB-INF/jsp/lesson01/ex02.jsp
```



## 컨트롤러 추가

- `@Controller` 만 붙은 클래스의 메소드에서 String을 리턴하면 String 경로의 view를 찾는다.

```
@Controller
public class Ex02Controller {
    @RequestMapping("/lesson01/ex02/1")
    public String ex02() {
        return "lesson01/ex02";
    }
}
```

## 요청해보기

```
http://localhost/lesson01/ex02
```

## Lombok 라이브러리 추가하기

### build.gradle에 lombok 의존성 라이브러리 추가

```
compileOnly 'org.projectlombok:lombok'
annotationProcessor 'org.projectlombok:lombok'
```

반드시 `build gradle Refresh` 한다.

## Lombok 라이브러리 STS와 연결

- [lombok.jar 다운로드](#)

### Windows

※ `lombok.jar` 의 위치가 한글 폴더로 되어있으면 STS가 안 열리므로 주의!!!

- `lombok.jar`를 수행시킨 후 `Specify location` 버튼으로 `sts.exe` 가 있는 위치를 로드 후 `install/update`

### MAC

- `lombok.jar`를 수행시킨 후 창이 뜨면 `Specify location` 버튼 클릭
- `SpringToolSuite4 > Contents > Eclipse > SpringToolSuite4.ini` 선택 후 `install/update`

### ini 파일에 추가 되어 있어야 함

```
-javaagent:D:\dulumary\spring\ex\sts\lombok.jar
```

## STS에서 Project > Clean 으로 빌드를 다시 한다.

### 동작 테스트

DTO 객체에서 `@Data` 어노테이션을 붙였을 때 에러가 나지 않고 `getter/setter`를 사용할 수 있으면 된다.

## JPA 설정

### build.gradle에 JPA 의존성 라이브러리 추가

```
implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
```

반드시 `build gradle` Refresh 한다.

## 쿼리 결과 보기(바인딩 된 파라미터) - p6spy 라이브러리 의존성 추가

```
implementation 'com.github.gavlyukovskiy:p6spy-spring-boot-starter:1.9.0'
```

반드시 `build gradle` Refresh 한다.

## 프로퍼티 설정

`application.properties` 에 설정을 추가한다.

key	설명
<code>spring.jpa.show-sql</code>	<code>true</code> : JPA 쿼리문 확인 가능
<code>spring.jpa.hibernate.ddl-auto</code>	<code>create</code> : 기존 테이블을 삭제 후 새로 생성 <code>create-drop</code> : 기존 테이블을 삭제 후 새로 생성 + 종료 후 다시 drop (drop + create + drop) <code>update</code> : 테이블과 매핑 정보 비교 후 변경사항만 수정(만약 기존 테이블 없으면 create) <code>validate</code> : 차이가 있으면 경고를 날린 후 수행하지 않음 <code>none</code> : 테이블 자동 생성기능 사용 안함(있는 테이블로 사용)
<code>spring.jpa.properties.hibernate.format_sql</code>	<code>true</code> : jpa 구현체인 하이버네이트 동적쿼리 로그를 예쁘게 출력

`application.properties` 파일에 아래 내용 추가

```
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=none
spring.jpa.properties.hibernate.format_sql=true
```

## 테이블 컬럼명을 CamelCase로 지정하는 설정

`application.properties` 파일에 아래 내용 추가

```
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```