# Spring Boot

# 스프링부트

https://spring.io/projects/spring-boot

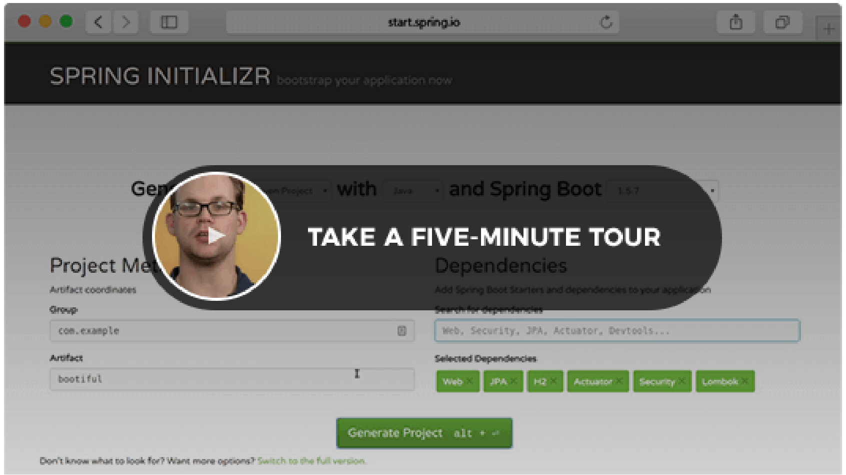
# Spring Boot

Spring Boot

## BUILD ANYTHING WITH SPRING BOOT

Spring Boot is the starting point for building all Spring-based applications. Spring Boot is designed to get you up and running as quickly as possible, with minimal upfront configuration of Spring.

- Get started in seconds using Spring Initializr
- Build anything: REST API, WebSocket, web, streaming, tasks, and more
- Simplified security
- Rich support for SQL and NoSQL
- Embedded runtime support: Tomcat, Jetty, and Undertow
- Developer productivity tools such as LiveReload and Auto Restart
- Curated dependencies that just work
- Production-ready features such as tracing, metrics, and health status
- Works in your favorite IDE: Spring Tool Suite, IntelliJ IDEA, and NetBeans



SPRING INITIALIZR bootstrap your application now

Gen... with ... and Spring Boot

Project Me...

Artifact coordinates

Group: com.example

Artifact: bootiful

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies: Web, Security, JPA, Actuator, Devtools...

Selected Dependencies: Web, JPA, H2, Actuator, Security, Lombok

Don't know what to look for? Want more options? Switch to the full version.

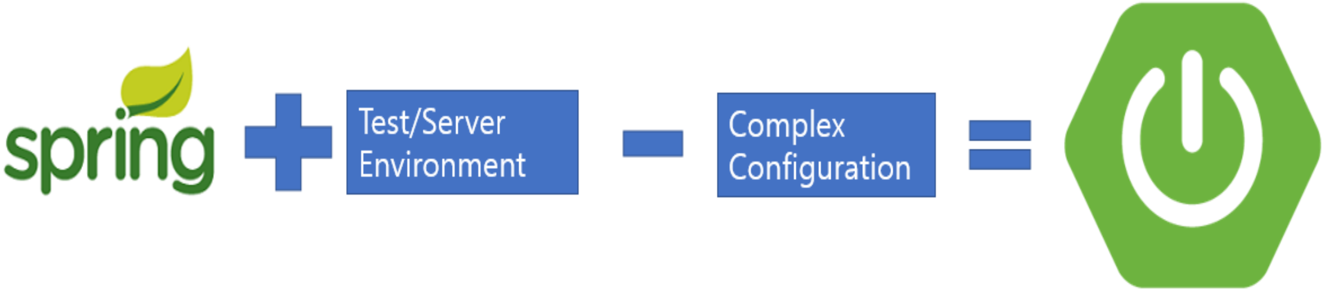
https://spring.io/

# Spring Boot

- **단독으로 실행 가능하고 제품 수준의 스프링기반 어플리케이션을 제작하는 것을 목표로 진행된 프로젝트**
- **주요기능**
  - 단독실행 가능한 수준의 스프링 애플리케이션 개발 가능
  - 내장된 WAS(예. Tomcat)를 이용해서 별도의 서버 설치 없이 실행 가능
  - 최대한 자동화된 설정
  - XML 설정 없이 단순한 설정 방식 제공

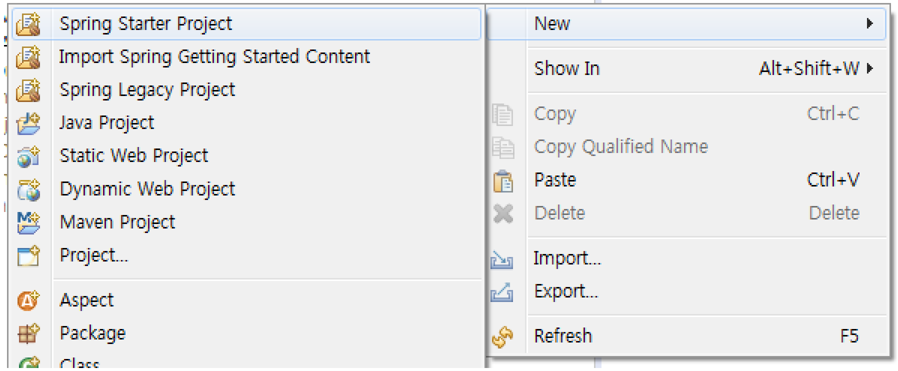
# Spring Boot

- **Spring Boot가 해결하는 것들**
  - 자동화된 라이브러리 관리
  - Spring Boot Auto Configuration
  - 라이브러리의 자동 결정과 XML없는 환경
  - 테스트 환경과 내장 Tomcat



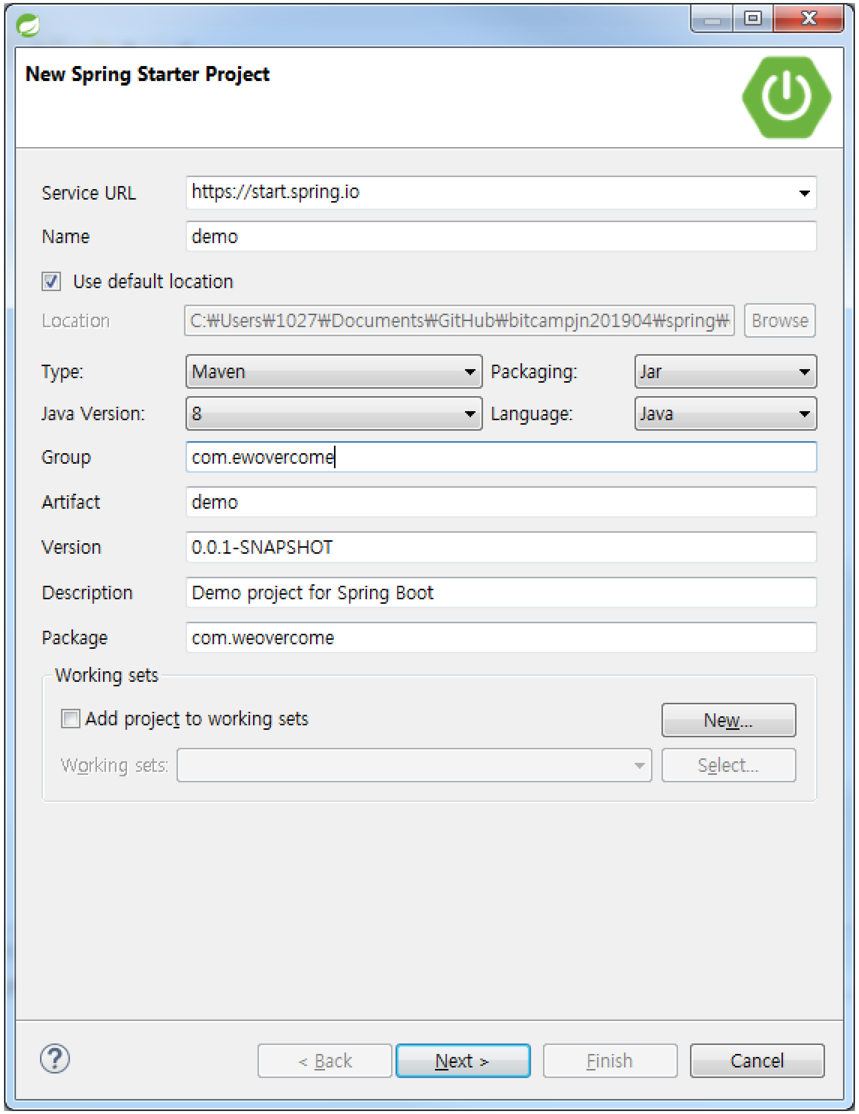
# Spring Boot

- **Spring Boot Project Start**



# Spring Boot

- **Spring Boot Project Start**



New Spring Starter Project

Service URL: https://start.spring.io

Name: demo

☑ Use default location

Location: C:₩Users₩1027₩Documents₩GitHub₩bitcampjn201904₩spring₩

Type: Maven

Packaging: Jar

Java Version: 8

Language: Java

Group: com.ewovercome

Artifact: demo

Version: 0.0.1-SNAPSHOT

Description: Demo project for Spring Boot

Package: com.weovercome

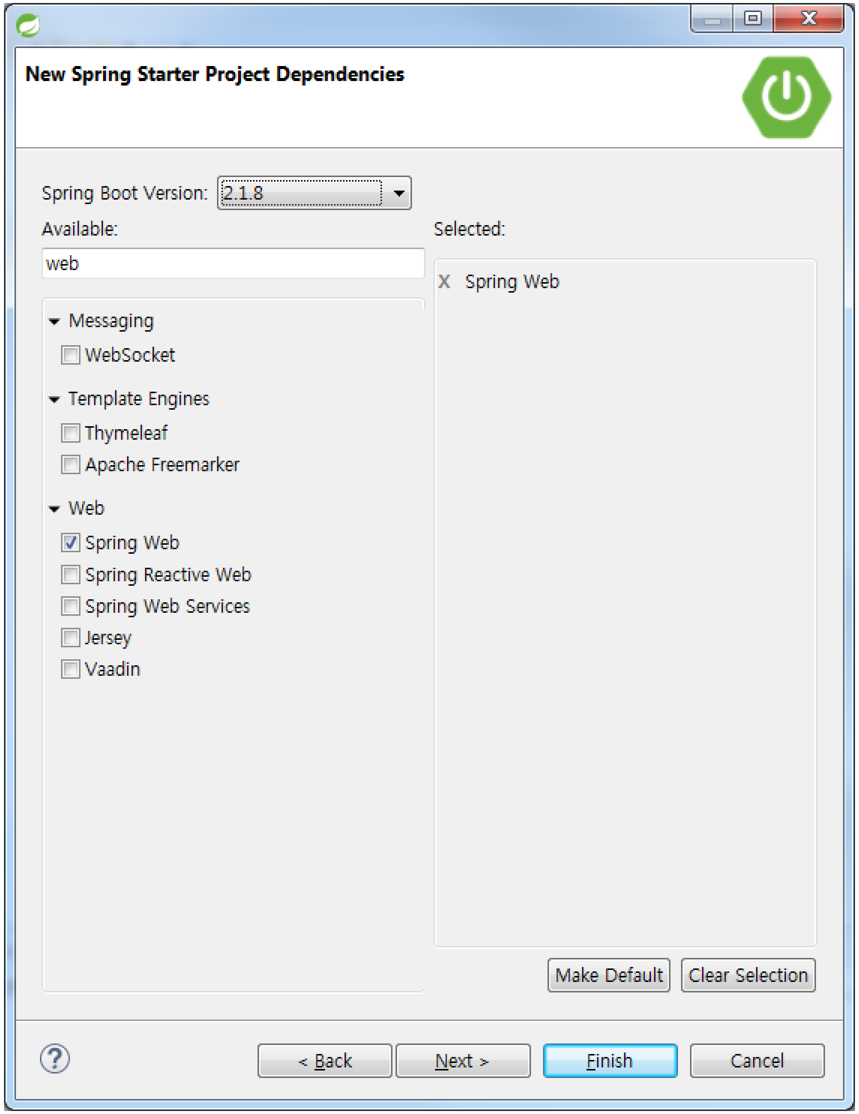
Working sets

□ Add project to working sets

Working sets:

# Spring Boot

- ## Spring Boot Project Start



New Spring Starter Project Dependencies

Spring Boot Version: 2.1.8

Available: web

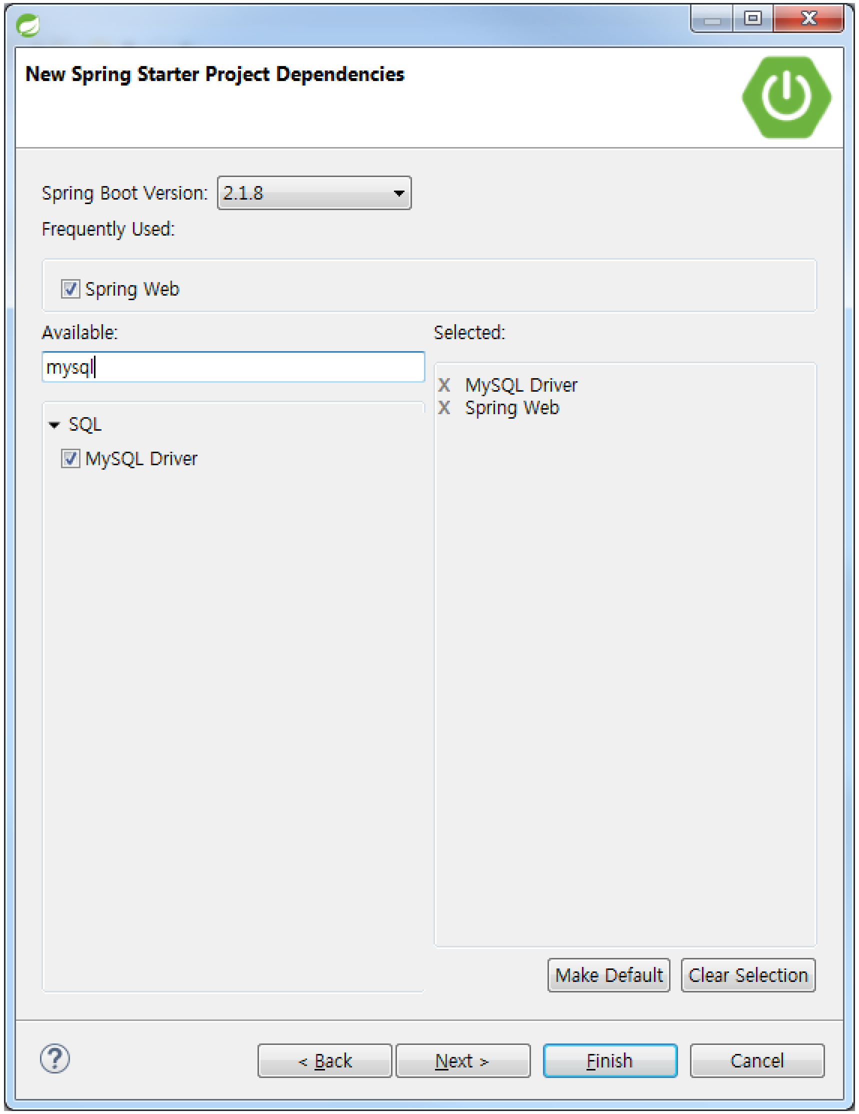
Selected:

X Spring Web

- Messaging
  - □ WebSocket
- Template Engines
  - □ Thymeleaf
  - □ Apache Freemarker
- Web
  - ☑ Spring Web
  - □ Spring Reactive Web
  - □ Spring Web Services
  - □ Jersey
  - □ Vaadin

# Spring Boot

- **Spring Boot Project Start**



**New Spring Starter Project Dependencies**

Spring Boot Version: 2.1.8

Frequently Used:

☑ Spring Web

Available: mysql

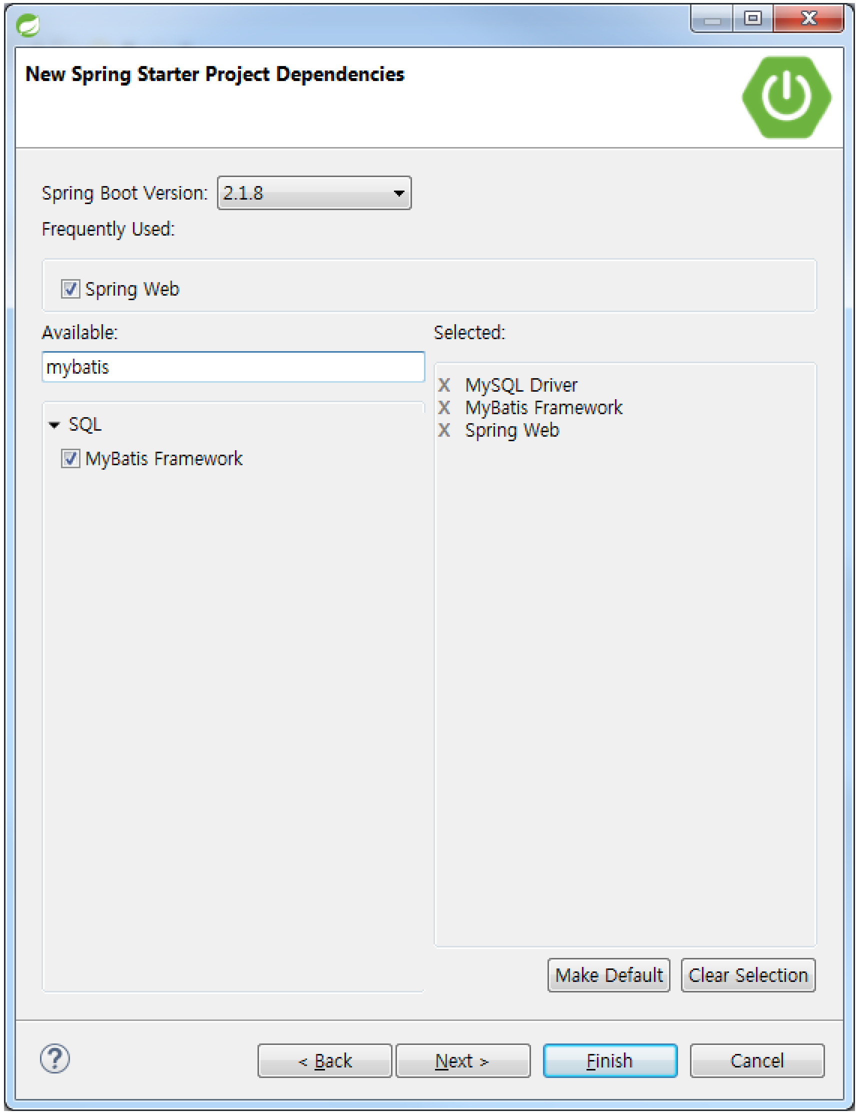
- SQL
  - ☑ MySQL Driver

Selected:

X MySQL Driver
X Spring Web

# Spring Boot

- ## Spring Boot Project Start



New Spring Starter Project Dependencies

Spring Boot Version: 2.1.8

Frequently Used:

☑ Spring Web

Available: mybatis

SQL

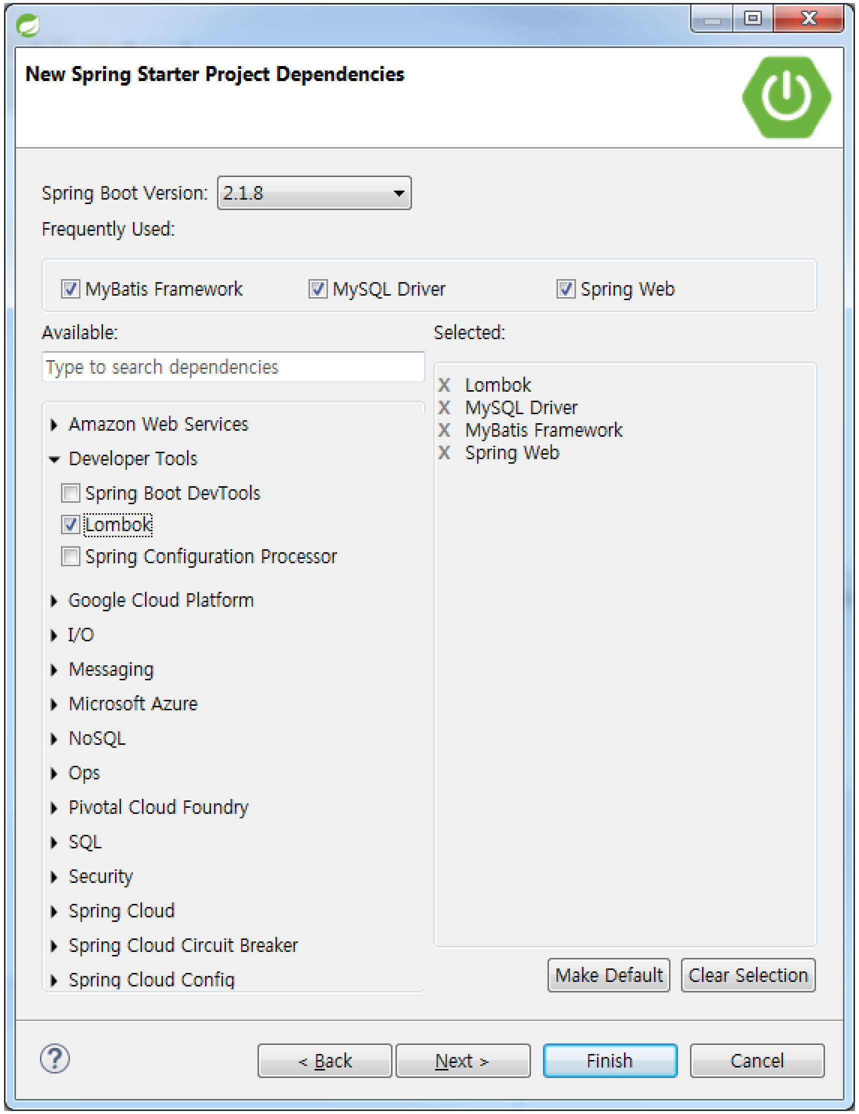
☑ MyBatis Framework

Selected:

X MySQL Driver
X MyBatis Framework
X Spring Web

# Spring Boot

- **Spring Boot Project Start**



New Spring Starter Project Dependencies

Spring Boot Version: 2.1.8

Frequently Used:

☑ MyBatis Framework ☑ MySQL Driver ☑ Spring Web

Available:

Type to search dependencies

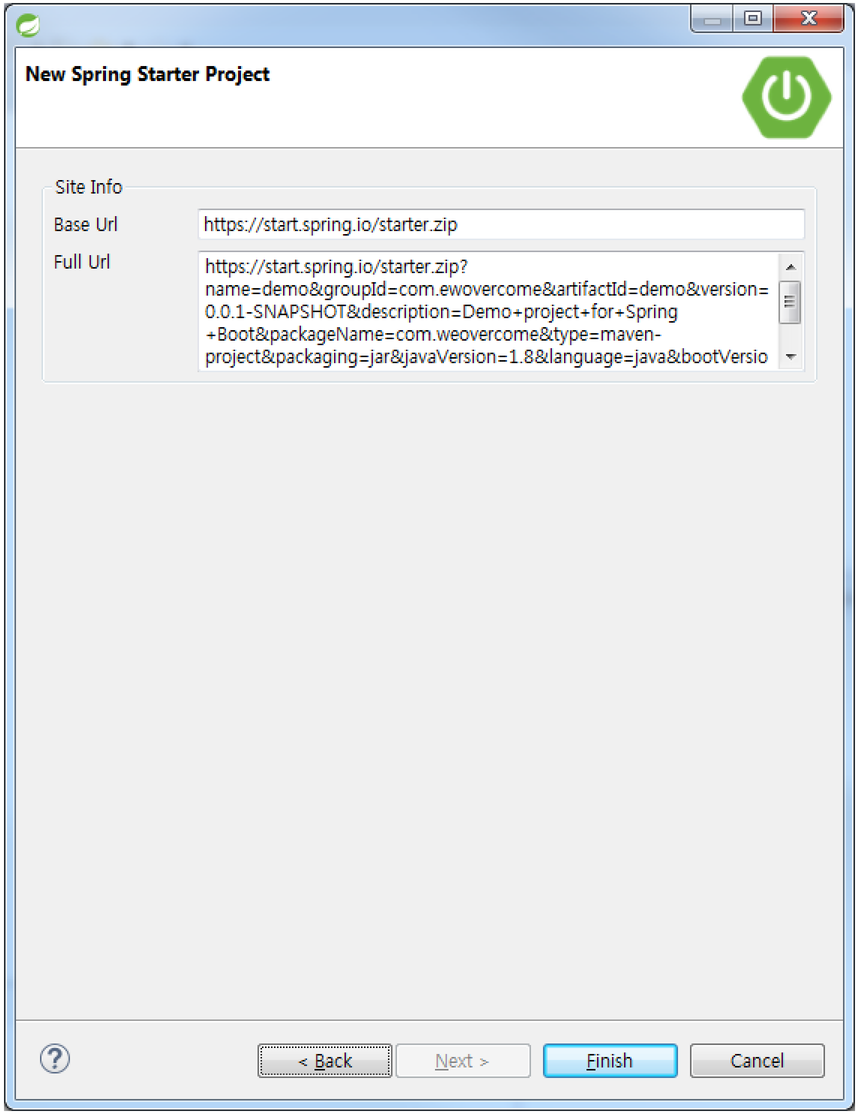
- Amazon Web Services
- Developer Tools
  - □ Spring Boot DevTools
  - ☑ Lombok
  - □ Spring Configuration Processor
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Ops
- Pivotal Cloud Foundry
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config

Selected:

X Lombok
X MySQL Driver
X MyBatis Framework
X Spring Web

# Spring Boot

- **Spring Boot Project Start**



New Spring Starter Project

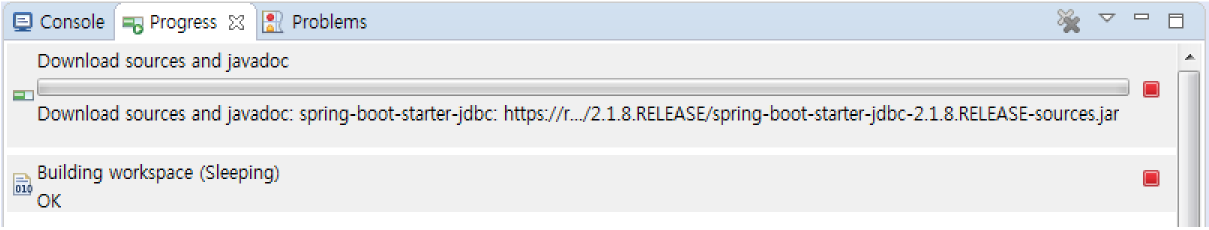
Site Info

Base Url: https://start.spring.io/starter.zip

Full Url: https://start.spring.io/starter.zip?name=demo&groupId=com.ewovercome&artifactId=demo&version=0.0.1-SNAPSHOT&description=Demo+project+for+Spring+Boot&packageName=com.weovercome&type=maven-project&packaging=jar&javaVersion=1.8&language=java&bootVersio

# Spring Boot

- **Spring Boot Project Start**



Download sources and javadoc

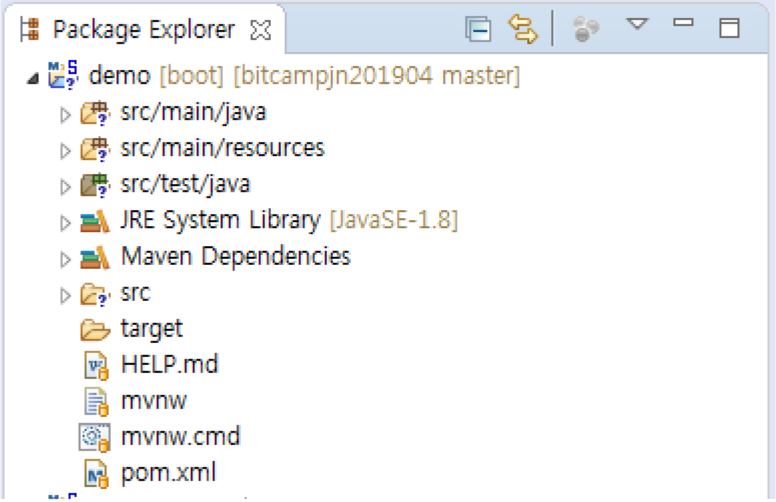
Download sources and javadoc: spring-boot-starter-jdbc: https://r.../2.1.8.RELEASE/spring-boot-starter-jdbc-2.1.8.RELEASE-sources.jar

Building workspace (Sleeping)

OK

# Spring Boot

- **Spring Boot Project Start**

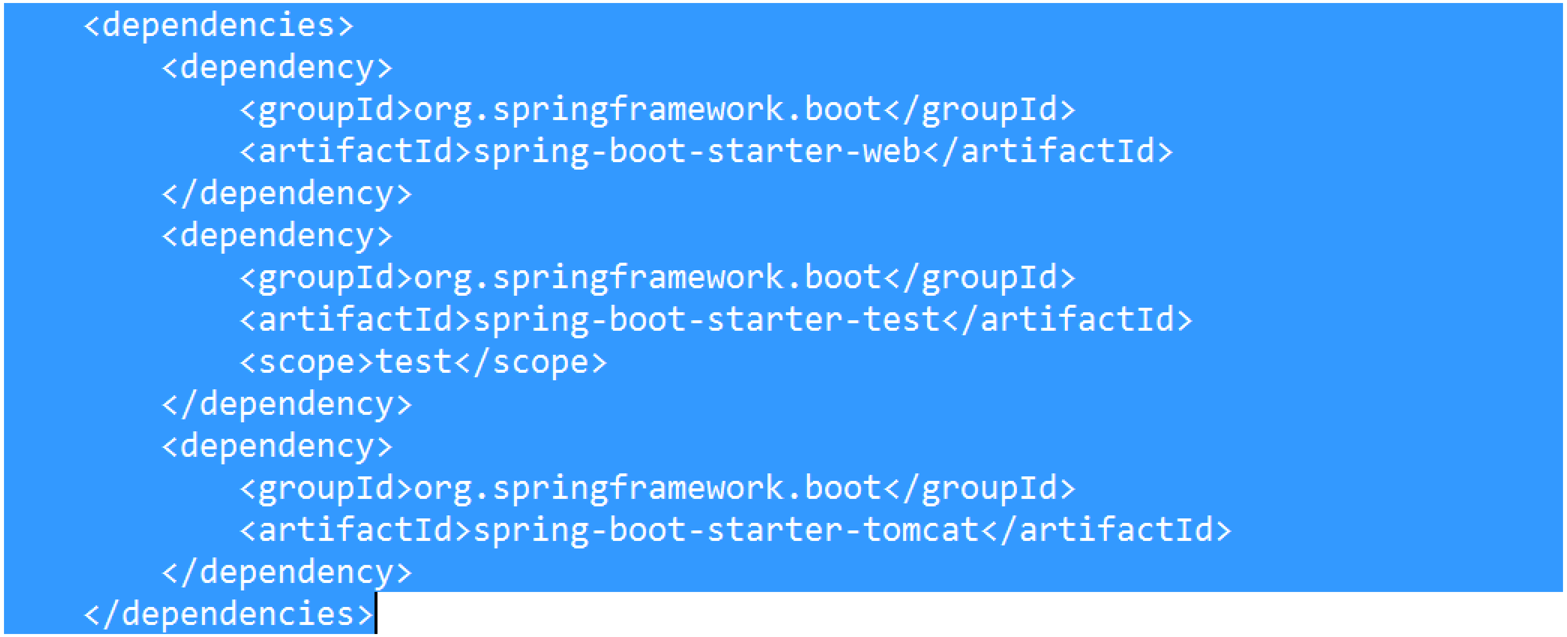


Package Explorer

- demo [boot] [bitcampjn201904 master]
  - src/main/java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

# Spring Boot

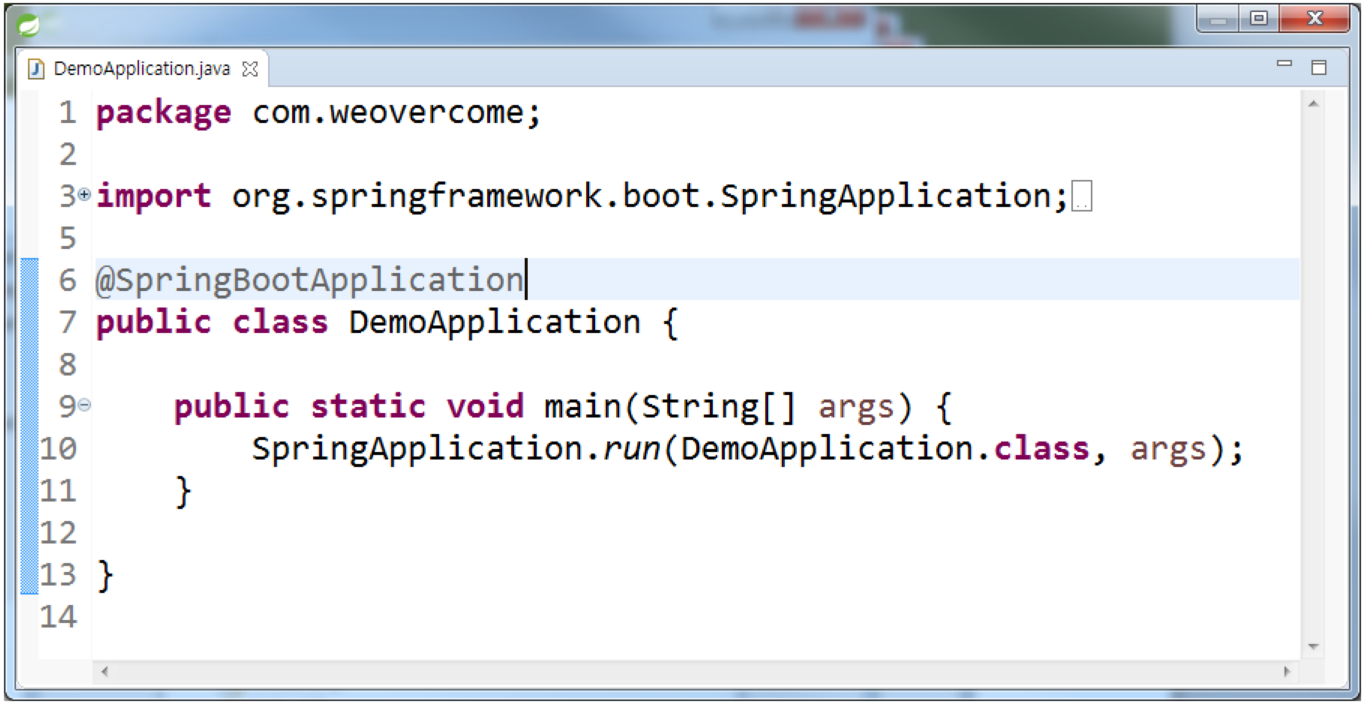
- **Spring Boot Project Start**



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

# Spring Boot

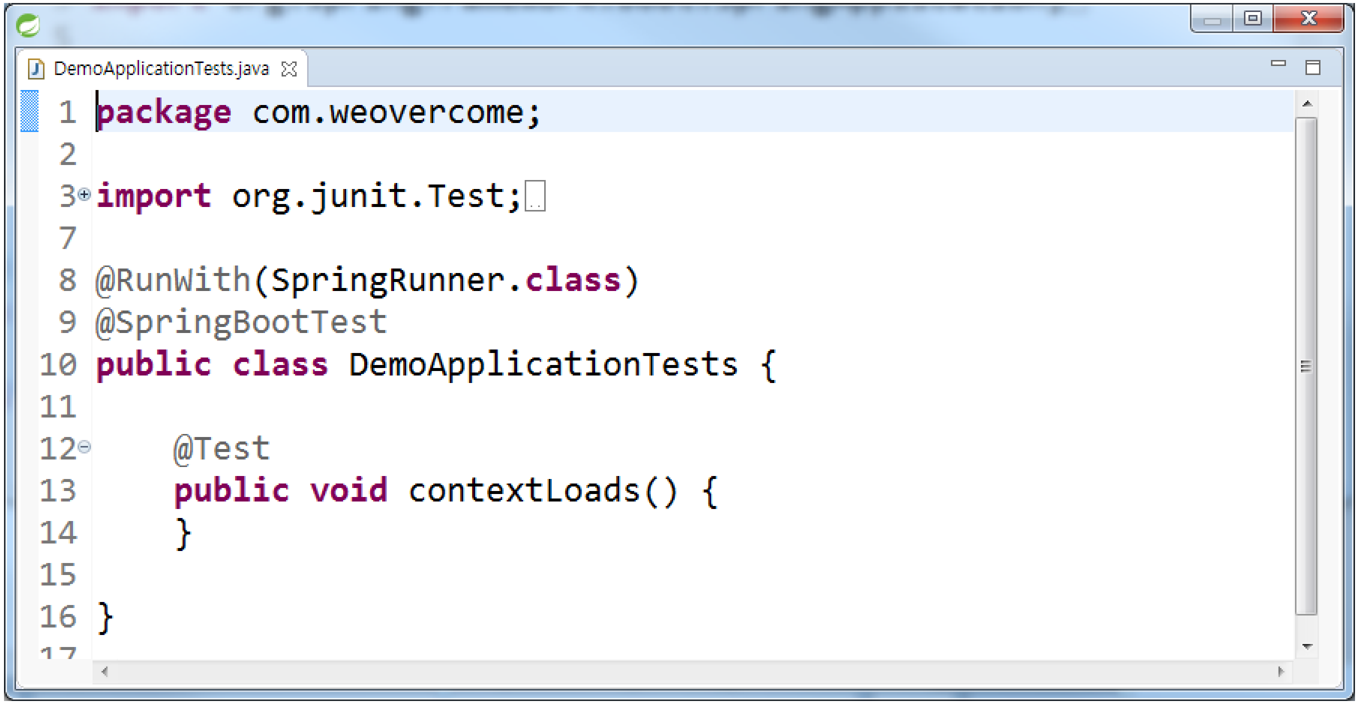
- **Spring Boot Project Start**



```java
package com.weovercome;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}

```

# Spring Boot

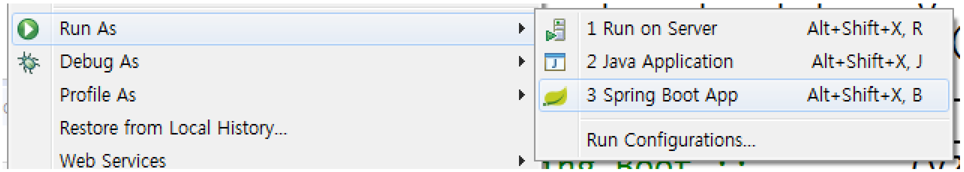
- **Spring Boot Project Start**

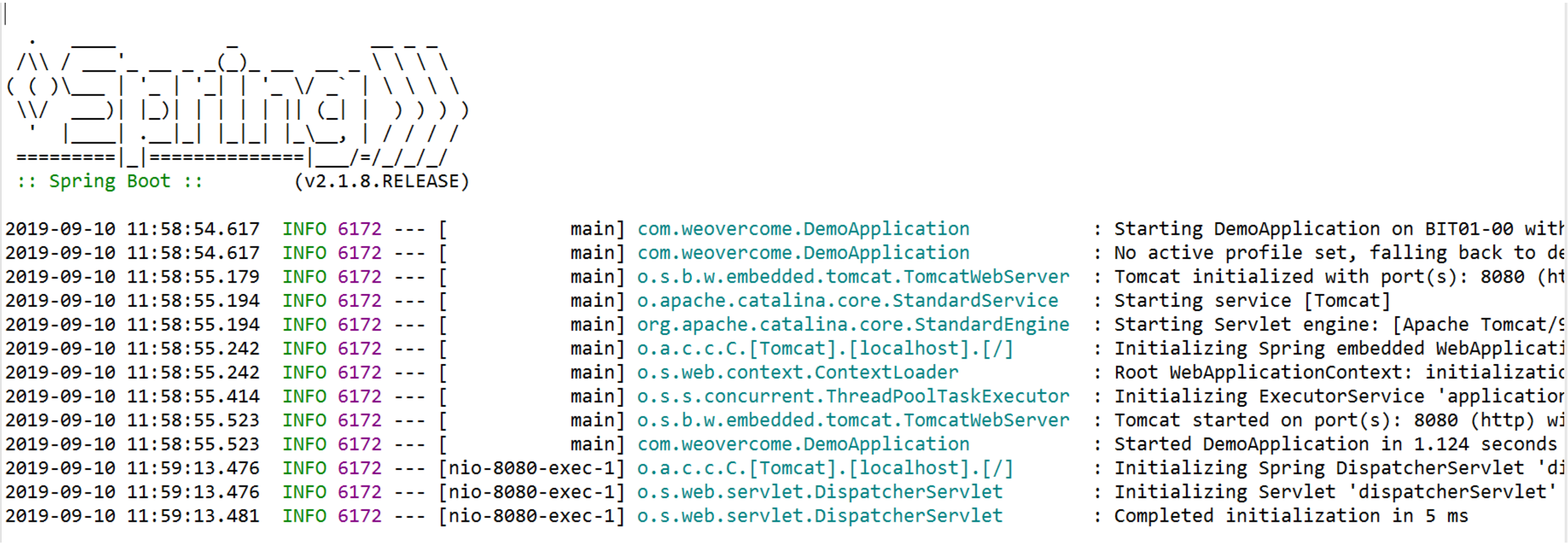


```java
package com.weovercome;

import org.junit.Test;

@RunWith(SpringRunner.class)
@SpringBootTest
public class DemoApplicationTests {

    @Test
    public void contextLoads() {
    }

}
```

# Spring Boot

- **Spring Boot Project Start**


Run As
Debug As
Profile As
Restore from Local History...
Web Services
1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
3 Spring Boot App Alt+Shift+X, B
Run Configurations...



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.8.RELEASE)

2019-09-10 11:58:54.617  INFO 6172 --- [           main] com.weovercome.DemoApplication           : Starting DemoApplication on BIT01-00 with
2019-09-10 11:58:54.617  INFO 6172 --- [           main] com.weovercome.DemoApplication           : No active profile set, falling back to de
2019-09-10 11:58:55.179  INFO 6172 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (ht
2019-09-10 11:58:55.194  INFO 6172 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2019-09-10 11:58:55.194  INFO 6172 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9
2019-09-10 11:58:55.242  INFO 6172 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicati
2019-09-10 11:58:55.242  INFO 6172 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initializatio
2019-09-10 11:58:55.414  INFO 6172 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'application
2019-09-10 11:58:55.523  INFO 6172 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) wi
2019-09-10 11:58:55.523  INFO 6172 --- [           main] com.weovercome.DemoApplication           : Started DemoApplication in 1.124 seconds
2019-09-10 11:59:13.476  INFO 6172 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'di
2019-09-10 11:59:13.476  INFO 6172 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2019-09-10 11:59:13.481  INFO 6172 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms
```

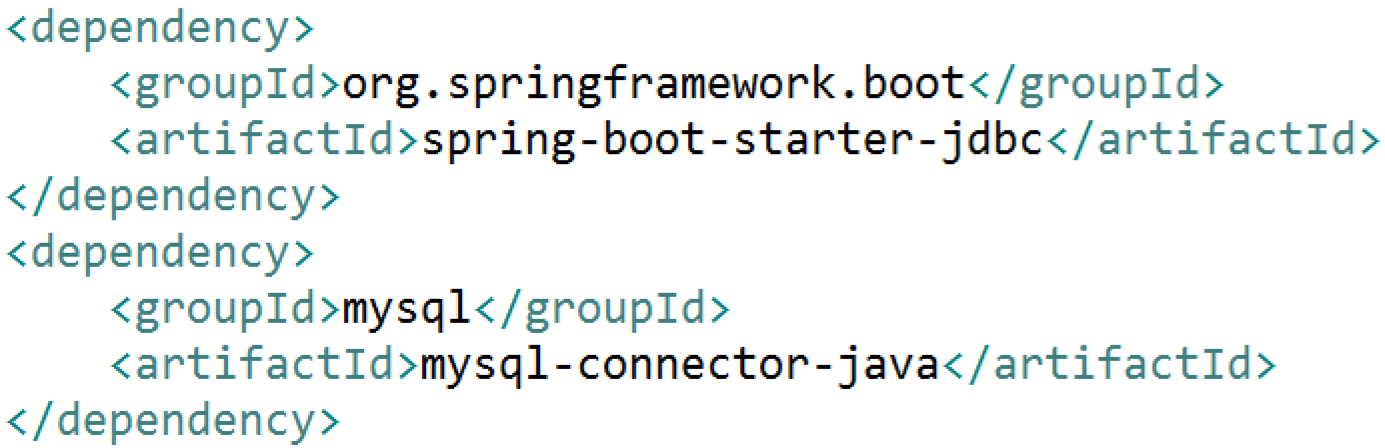
# Spring Boot

# +

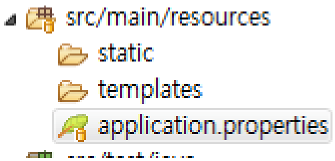
# Mysql

# Spring Boot

- static : HTML, CSS, JavaScript, 이미지 파일들을 보관하는 경로
- templates : Thymeleaf와 같은 템플릿 경로
- application.properties : 에플리케이션 내의 설정파일



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
</dependency>
```



```
src/main/resources
    static
    templates
    application.properties
```

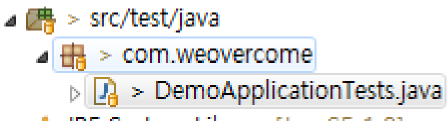
# Spring Boot

- **DataSource 설정**
  - @Bean 을 이용한 DataSource 설정
  - application.properties 를 이용한 데이터 설정

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/project?autoReconnect=true&useSSL=false&serverTimezone=UTC
spring.datasource.username=bit
spring.datasource.password=bit
```

# Spring Boot

- datasource 테스트



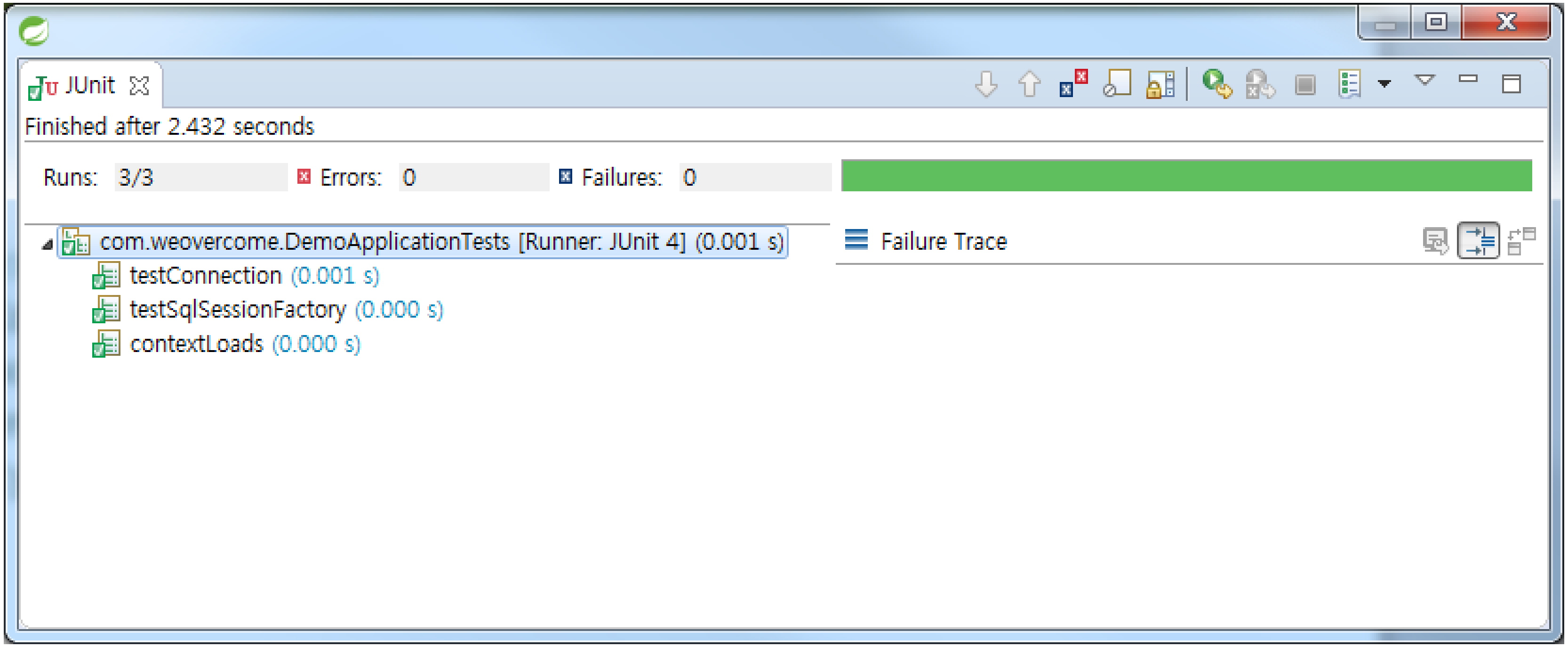
src/test/java
com.weovercome
DemoApplicationTests.java

```java
package com.weovercome;

import java.sql.Connection;
import java.sql.SQLException;

import javax.sql.DataSource;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

@RunWith(SpringRunner.class)
@SpringBootTest
public class DemoApplicationTests {

            @Autowired
            private DataSource datasource;

            @Test
            public void contextLoads() {
            }

            @Test
            public void testConnection() throws SQLException {
                        System.out.println(datasource);
                        Connection conn = datasource.getConnection();
                        System.out.println(conn);
                        conn.close();
            }

}
```

# Spring Boot

- **Mybatis 설정**

```
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>2.1.0</version>
</dependency>
```



Finished after 2.432 seconds

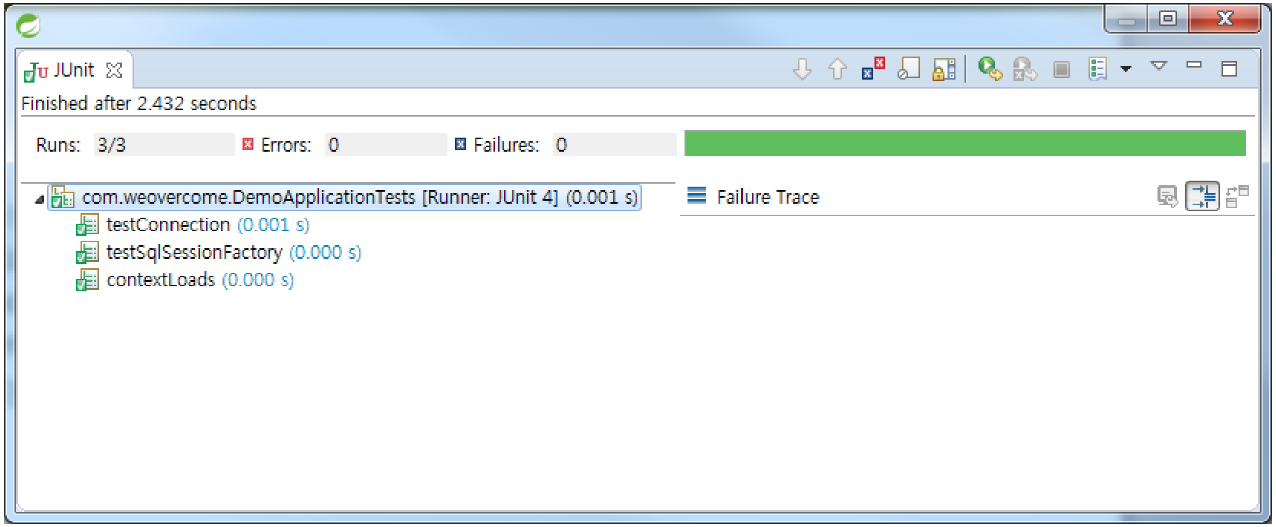
Runs: 3/3 Errors: 0 Failures: 0

```
com.weovercome.DemoApplicationTests [Runner: JUnit 4] (0.001 s)
    testConnection (0.001 s)
    testSqlSessionFactory (0.000 s)
    contextLoads (0.000 s)
```

# Spring Boot

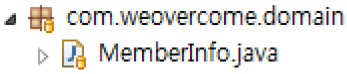
- **Mybatis 설정**

```
@Autowired
private SqlSessionFactory sessionFactory;

@Test
public void testSqlSessionFactory() {
        System.out.println(sessionFactory);
}
```



```
Finished after 2.432 seconds

Runs: 3/3    Errors: 0    Failures: 0

com.weovercome.DemoApplicationTests [Runner: JUnit 4] (0.001 s)
    testConnection (0.001 s)
    testSqlSessionFactory (0.000 s)
    contextLoads (0.000 s)
```

- **MemberInfo.java**

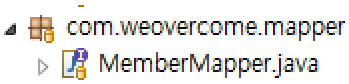


com.weovercome.domain
MemberInfo.java

```
package com.weovercome.domain;

public class MemberInfo {

        private int idx;
        private String uId;
        private String uPW;
        private String uName;
        private String uPhoto;
        private Date date;
        private char verify;
        private String code;

        // default 생성자 필수
        public MemberInfo() {
                this.date = new Date();
                getRandomSting();
        }

        // 변수들의 Getter/Setter 시작
}
```

# Spring Boot

- **Mapper 생성**

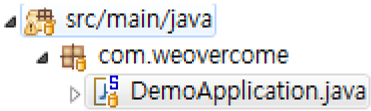


com.weovercome.mapper
MemberMapper.java

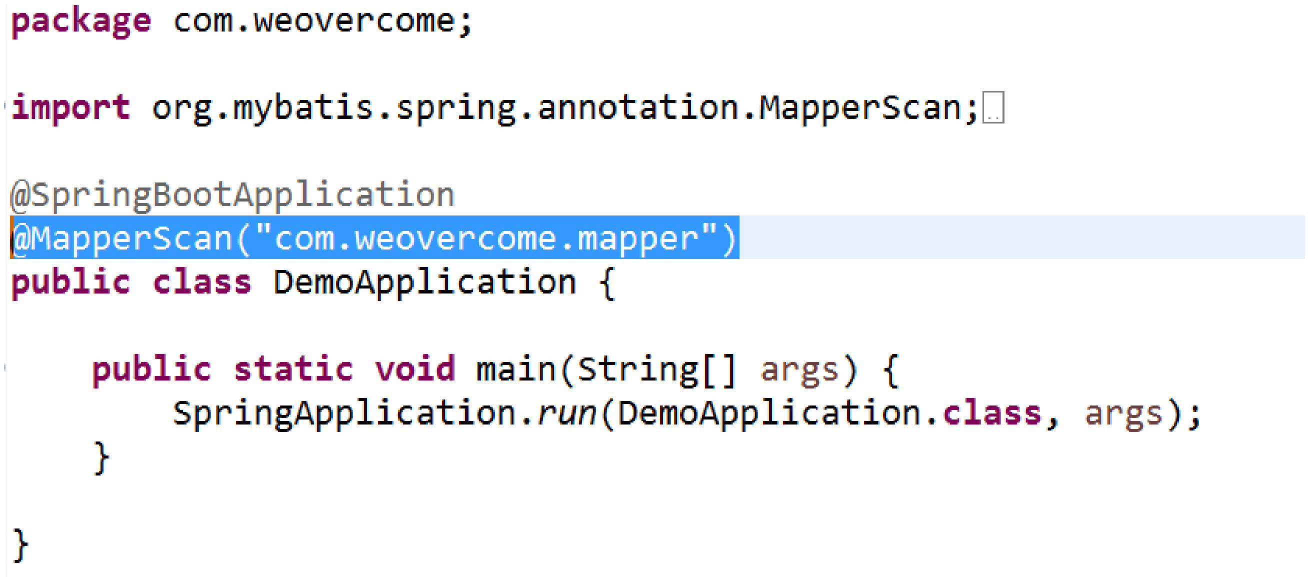
```
package com.weovercome.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.springframework.web.bind.annotation.RequestParam;

import com.weovercome.domain.MemberInfo;

public interface MemberMapper {

         @Select("SELECT * FROM member order by idx")
         public List<MemberInfo> getMemberList();
         public MemberInfo selectMemberById(
                  @RequestParam("uId") String uId);
 }
```

# Spring Boot

- **Mapper 설정**



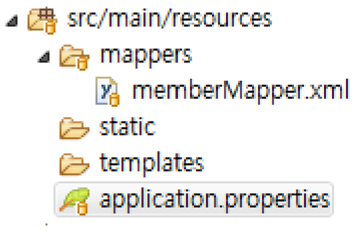
src/main/java
com.weovercome
DemoApplication.java



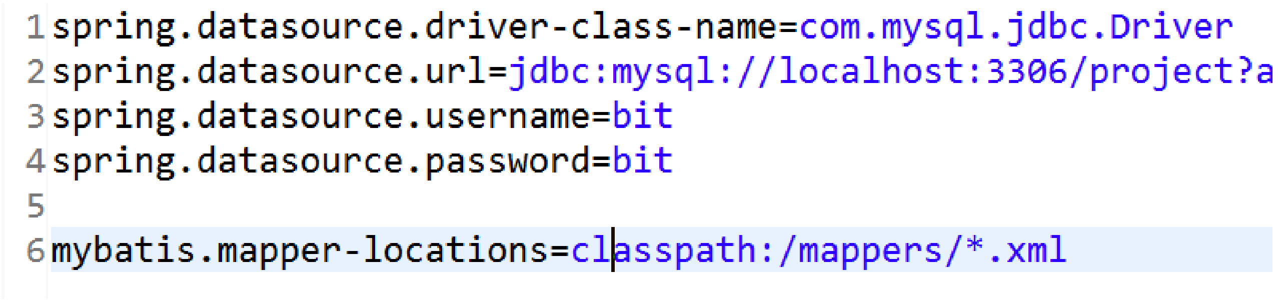
```
package com.weovercome;

import org.mybatis.spring.annotation.MapperScan;

@SpringBootApplication
@MapperScan("com.weovercome.mapper")
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

# Spring Boot

- **외부 XML mapper 설정**



```
src/main/resources
  mappers
    memberMapper.xml
  static
  templates
  application.properties
```



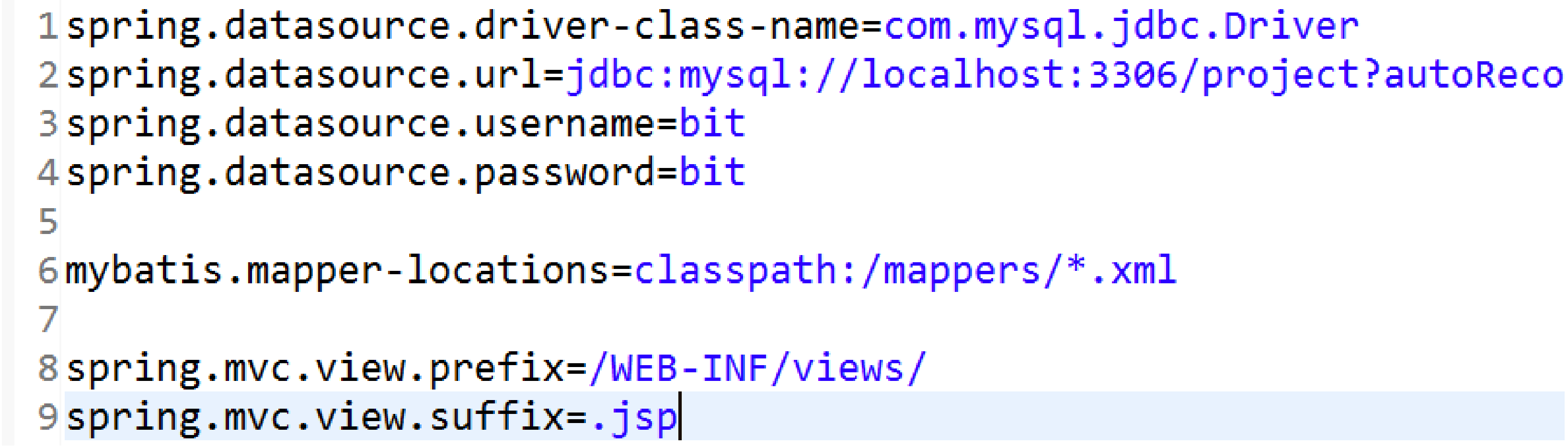
```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/project?a
spring.datasource.username=bit
spring.datasource.password=bit

mybatis.mapper-locations=classpath:/mappers/*.xml
```

# Spring Boot

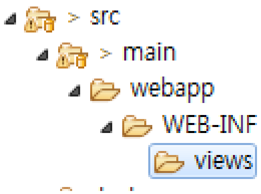
- **View 처리 방법**
  - **JSP 사용방법**
    - **apllication.properties 설정 변경**
    - **Tomcat 서버의 JSP 설정 변경**
  - **Thymeleaf 사용방법**

# Spring Boot

- **JSP사용 설정**
  - apllication.properties 설정 변경



```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/project?autoReco
spring.datasource.username=bit
spring.datasource.password=bit

mybatis.mapper-locations=classpath:/mappers/*.xml

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



경로에 맞게 폴더 생성해야 한다.
main > webapp > WEB-INF > views

# Spring Boot

- **JSP사용 설정**
  - Tomcat 서버의 JSP 설정 변경

```xml
<dependency>
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
</dependency>

<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
</dependency>
```

# Spring Boot

- **Github**
  - https://github.com/ryuyj/bitcampjn201904/tree/master/spring/demo