

Spring 프로젝트 생성

프로젝트 생성

File - new - Spring Legacy Project

New Spring Legacy Project

Spring Legacy Project

Enter a project name.

Project name:

☒ Use default location

Location:

Select Spring version:

Templates:

- Simple Projects
 - Simple Java
 - Simple Spring Maven
 - Simple Spring Web Maven
- Batch
- GemFire
- Integration
- Persistence
- Simple Spring Utility Project (requires downloading)
- Spring MVC Project

Description:

Working sets

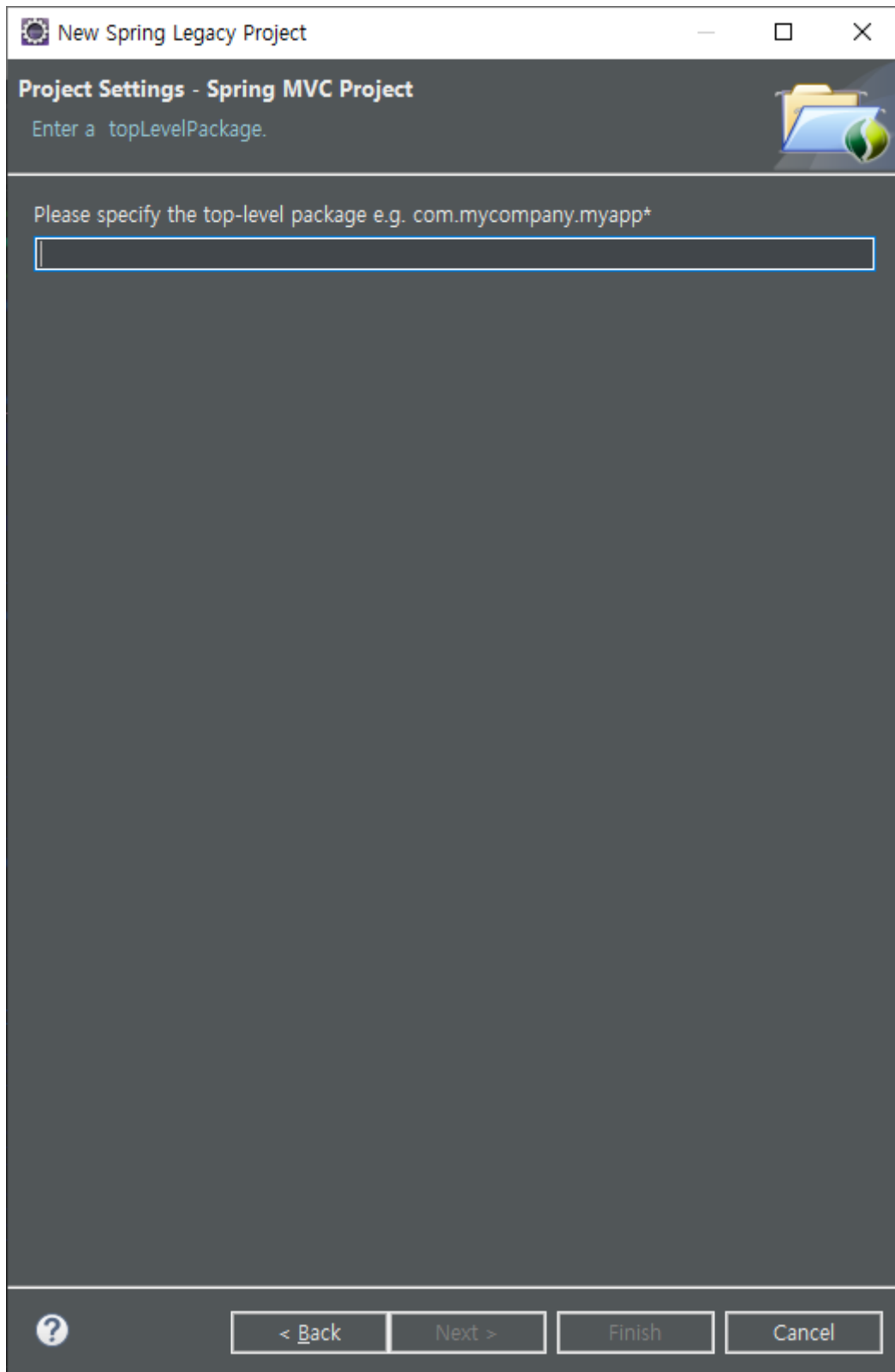
☐ Add project to working sets

Working sets:

Project name : 프로젝트명 입력

Templates : Spring MVC Project 선택

Next



The image shows a 'New Spring Legacy Project' dialog box. The title bar contains the text 'New Spring Legacy Project' and standard window controls. The main area has a dark background. At the top, it says 'Project Settings - Spring MVC Project' with a folder icon. Below this, it prompts 'Enter a topLevelPackage.' and 'Please specify the top-level package e.g. com.mycompany.myapp*'. A text input field is provided for this purpose. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >', 'Finish', and 'Cancel'.

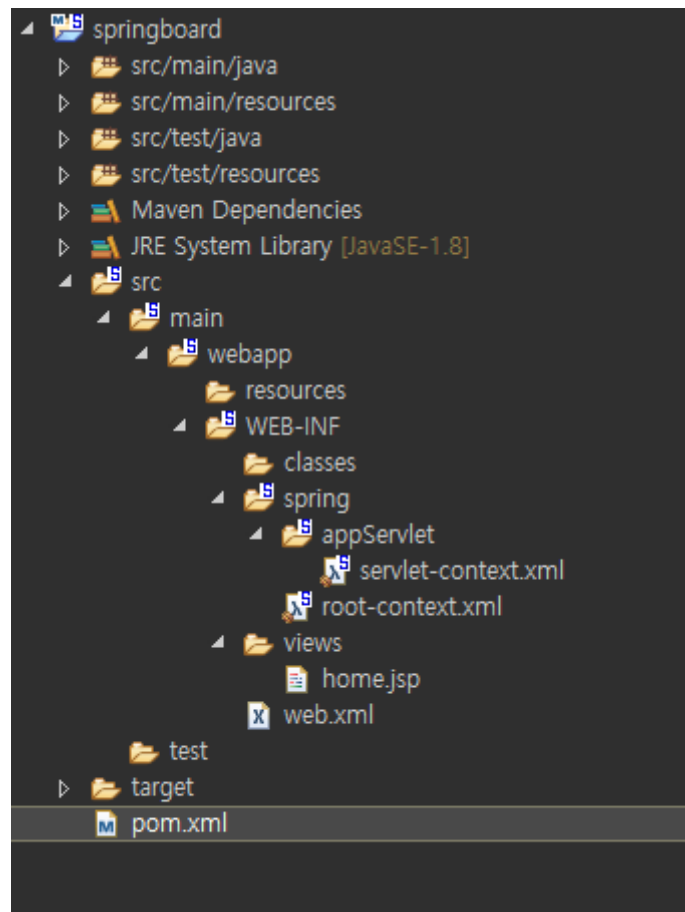
New Spring Legacy Project

Project Settings - Spring MVC Project

Enter a topLevelPackage.

Please specify the top-level package e.g. com.mycompany.myapp*

? < Back Next > Finish Cancel



각 파일들이 하는 일

src/main/java : 작성되는 코드의 경로

src/main/resources : 실행할 때 참고하는 기본 경로(주로 설정 파일들이 위치)

src/test/java : 테스트 코드의 경로

src/test/resources : 테스트 관련 설정 파일 보관 경로

JRE System Library : JavaSE - 1.6

servlet-context.xml : 웹과 관련된 스프링 설정 파일

root-context.xml : 스프링 설정 파일

views : 템플릿 프로젝트의 JSP파일 경로

web.xml : Tomcat의 web.xml 파일

pom.xml : 라이브러리를 관리하는 Maven이 사용하는 pom.xml

servlet-context.xml : 웹과 관련된 스프링 설정파일

root-context.xml : 스프링 설정파일

views : 템플릿 프로젝트의 JSP 파일 경로

web.xml : Tomcat 의 web.xml파일

pom.xml 수정

```

/pom.xml

<properties>
    <java-version>1.8</java-version>    # 버전 수정
    <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
# 버전 수정
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>

```

pom.xml 의 버전을 수정해준다.

spring-tx : 트랜잭션을 사용하기 위한 라이브러리

spring-jdbc : 데이터베이스와 연동하기 위한 라이브러리

spring-test : 테스트 코드를 단위 테스트하기 위한 라이브러리

```

/pom.xml

<!-- Spring -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${org.springframework-version}</version>
    <exclusions>
        <!-- Exclude Commons Logging in favor of SLF4j -->
        <exclusion>
            <groupId>commons-logging</groupId>
            <artifactId>commons-logging</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<!-- Spring Transaction -->
    # spring-tx : 트랜잭션을 사용하기 위한 라이브러리
<!-- Spring JDBC -->
    # spring-jdbc : 데이터베이스와 연동하기 위한 라이브러리
<!-- Spring Test -->
    # spring-test : 테스트 코드를 단위 테스트하기 위한 라이브러리
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${org.springframework-version}</version>
</dependency>

```

위에 새로 추가한 자리에 아래와 같이 추가

```

/pom.xml

<!-- Spring Transaction -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>${org.springframework-version}</version>
</dependency>

```

```

<!-- Spring JDBC -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>

<!-- Spring Test -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${org.springframework-version}</version>
</dependency>

```

MyBatis 관련 라이브러리 추가

HikariCP : 대용량 데이터 처리속도를 빠르게 하기 위한 라이브러리

MyBatis : 데이터베이스 프레임워크

mybatis-spring : MyBatis 를 사용하기 위한 스프링 라이브러리

Log4jdbc : 데이터베이스 실행 시, 단위 테스트를 위한 라이브러리

```

/pom.xml

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>

<!-- HikariCP -->
  # HikariCP : 대용량 데이터 처리속도를 빠르게 하기 위한 라이브러리
<!-- MyBatis -->
  # MyBatis : 데이터베이스 프레임워크
<!-- mybatis-spring -->
  # mybatis-spring : MyBatis 를 사용하기 위한 스프링 라이브러리
<!-- Log4jdbc -->
  # Log4jdbc : 데이터베이스 실행 시, 단위 테스트를 위한 라이브러리
<!-- AspectJ -->
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjrt</artifactId>
  <version>${org.aspectj-version}</version>
</dependency>

```

아래처럼 변경

```

/pom.xml

<!-- HikariCP -->
<dependency>
  <groupId>com.zaxxer</groupId>
  <artifactId>HikariCP</artifactId>

```

```

        <version>2.7.8</version>
    </dependency>

    <!-- MyBatis -->
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.4.6</version>
    </dependency>

    <!-- mybatis-spring -->
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>1.3.2</version>
    </dependency>

    <!-- Log4jdbc -->
    <dependency>
        <groupId>org.bgee.log4jdbc-log4j2</groupId>
        <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
        <version>1.16</version>
    </dependency>

```

lombok.jar 추가

설치방법(cmd)

```
>>> java -jar lombok.jar
```

```

/pom.xml

<!-- Test -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>    # 버전 변경
    <scope>test</scope>
</dependency>

<!-- lombok -->    # 추가
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.0</version>
    <scope>provided</scope>
</dependency>

</dependencies>
<build>
    <plugins>
        <plugin>
            <artifactId>maven-eclipse-plugin</artifactId>
            <version>2.9</version>

```

```

    <plugin>
      <artifactId>maven-eclipse-plugin</artifactId>
      <version>2.9</version>
      <configuration>
        <additionalProjectnatures>

    <projectnature>org.springframework.ide.eclipse.core.springnature</projectnature
  >
      </additionalProjectnatures>
      <additionalBuildcommands>

    <buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
      </additionalBuildcommands>
      <downloadSources>true</downloadSources>
      <downloadJavadocs>true</downloadJavadocs>
    </configuration>
  </plugin>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.5.1</version>
    <configuration>
      <source>1.8</source>      # 수정
      <target>1.8</target>      # 수정
      <compilerArgument>-Xlint:all</compilerArgument>
      <showWarnings>true</showWarnings>
      <showDeprecation>true</showDeprecation>
    </configuration>
  </plugin>

```

```

<!-- Servlet -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>      # 수정
  <version>3.1.0</version>      # 수정
  <scope>provided</scope>
</dependency>

```