

## 1. 스프링 프레임워크 시작하기



# 차례

---

- 프레임워크란?
- 스프링 프레임워크
- 스프링 프레임워크 환경 설정하기

# 1. 프레임워크란?

---

## ■ 프레임워크(Framework) 정의

- 사전적 의미는 ‘어떤 것을 구성하는 구조 또는 뼈대’
- 소프트웨어적 의미로는 ‘기능을 미리 클래스나 인터페이스 등으로 만들어 제공하는 반제품’

## ■ 프레임워크(Framework) 장점

- 일정한 기준에 따라 개발이 이루어지므로 개발 생산성과 품질이 보장된 애플리케이션을 개발할 수 있음
- 개발 후 유지보수 및 기능의 확장성에서도 고품질 보장

## 2. 스프링 프레임워크

---

### ■ 스프링 프레임워크

- 스프링 프레임워크(이하 스프링)는 자바 웹 애플리케이션 개발을 위한 오픈 소스 프레임워크
- EJB(Enterprise Java Bean, 엔터프라이즈 자바 빈즈)보다 가벼운 경량 프레임워크 (lightWeight Framework)

### ■ 컨테이너(Container)란?

- 톰캣은 서블릿 컨테이너라고 부르는데, 그 이유는 톰캣을 실행하면 톰캣은 서블릿의 생성, 초기화, 서비스 실행, 소멸에 관한 모든 권한을 가지고 서블릿을 관리
- 스프링은 애플리케이션에서 사용되는 여러 가지 빈(클래스 객체)을 개발자가 아닌 스프링이 권한을 가지고 직접 관리

## 2. 스프링 프레임워크

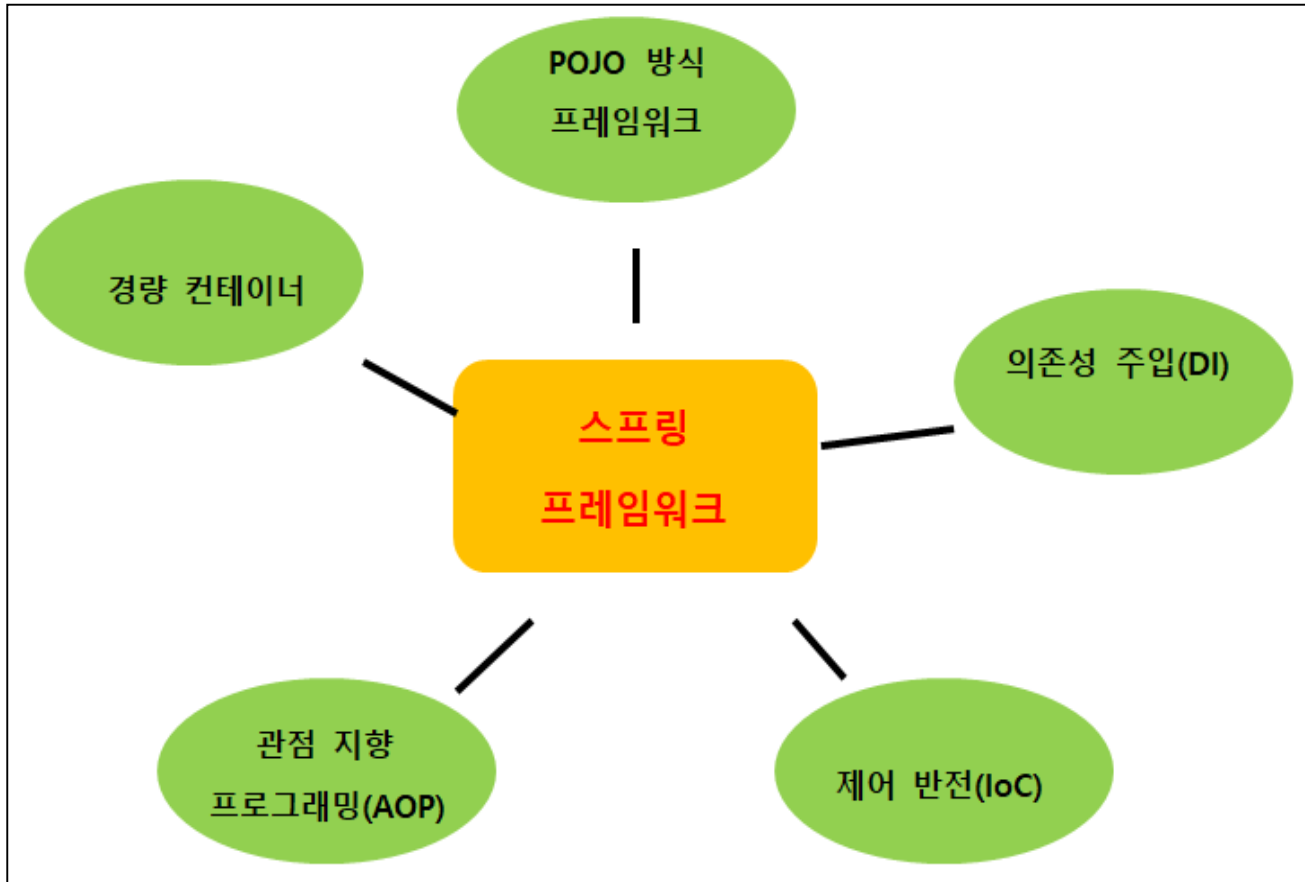
---

### ■ 스프링의 특징

- EJB보다 가볍고 배우기도 쉬우며 경량 컨테이너의 기능을 수행
- 제어 역행(LoC, Inversion of Control) 기술을 이용해 애플리케이션 간의 느슨한 결합을 제어함
- 의존성 주입(DI, Dependency Injection) 기능을 지원함
- 관점 지향 (AOP, Aspect-Oriented Programming) 기능을 이용해 자원 관리함
- 영속성과 관련된 다양한 서비스를 지원함
- 수많은 라이브러리와 연동 기능을 지원함

## 2. 스프링 프레임워크

### ■ 스프링 프레임워크의 특징



## 2. 스프링 프레임워크

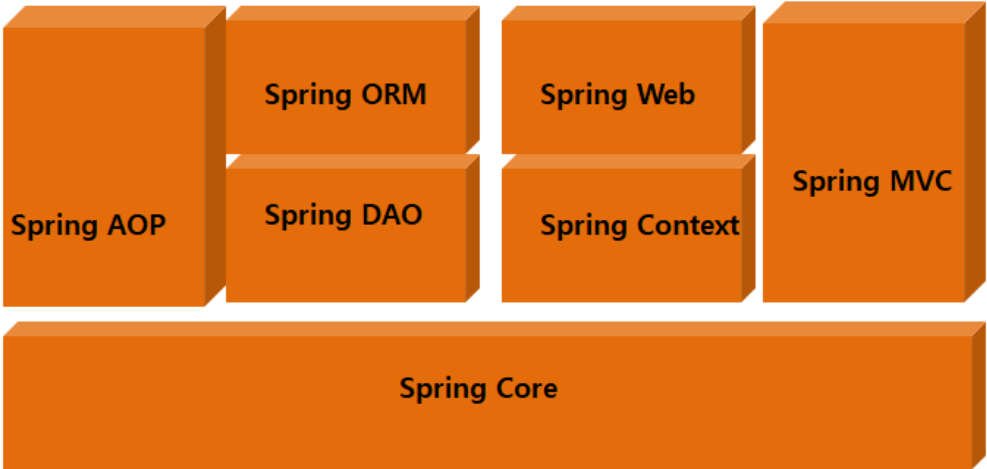
---

### ■ 용어 정리

- 의존성 주입: 클래스 객체를 개발자가 코드에서 생성하지 않고 프레임워크가 생성하여 사용하는 방법
- 제어 역행: 서블릿이나 빈 등을 개발자가 코드에서 생성하지 않고 프레임워크가 직접 수행하는 방법
- 관점 지향: 핵심 기능 외 부수 기능들을 분리 구현함으로써 모듈성을 증가시키는 방법

## 2. 스프링 프레임워크

- 스프링 프레임워크의 주요 기능

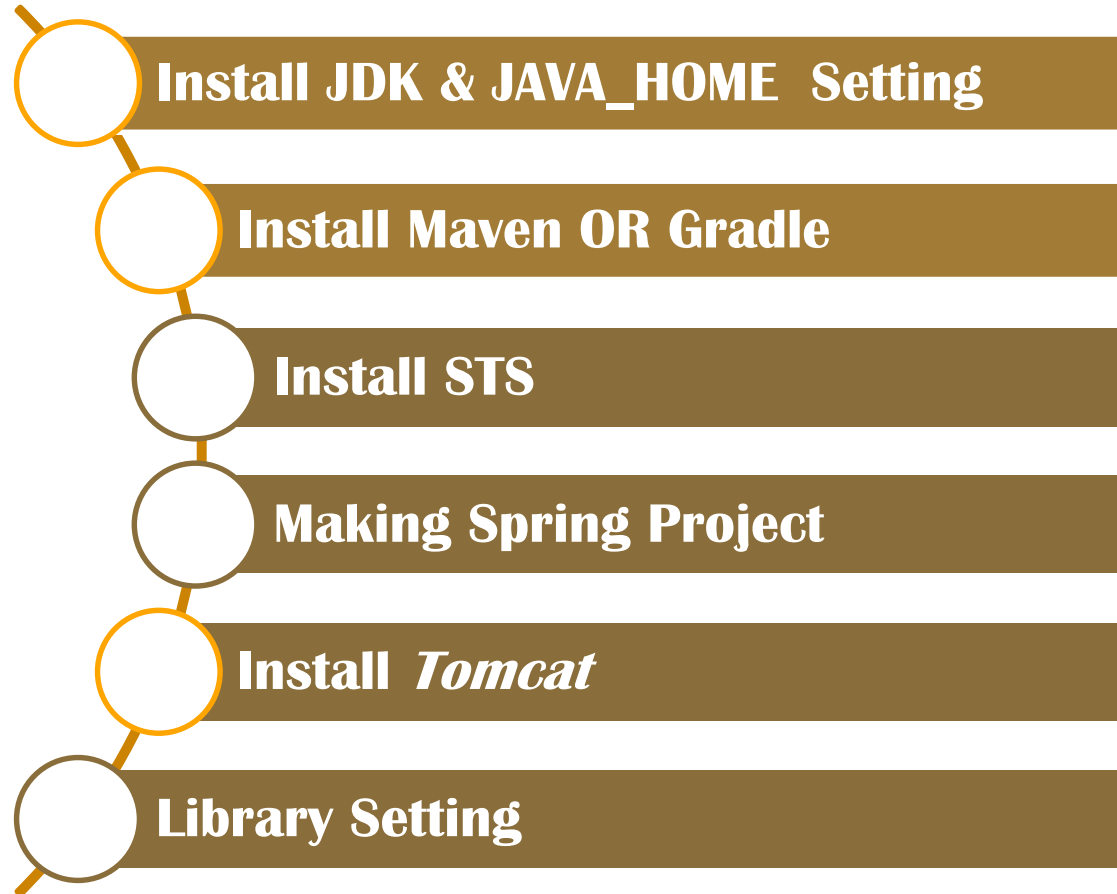


기능	설명
Core	다른 기능과 설정을 분리하기 위한 IoC 기능을 제공
Context	스프링의 기본 기능으로서 애플리케이션의 각 기능을 하는 빈(Beans)에 대한 접근 방법을 제공
DAO	JDBC 기능을 좀 더 편리하게 사용할 수 있도록 합니다.
ORM	하이버네이트나 마이바티스 같은 영속성 관련 프레임워크와 연동된 기능을 제공
AOP	관점 지향 기능을 제공
Web	웹 애플리케이션 개발에 필요한 기능을 제공
WebMVC	스프링에서 MVC 구현에 관련된 기능을 제공



### 3. 스프링 프레임워크 환경설정

---



# 개발 환경의 우선순위

---

Spring 프레임워크의 설치

개발용 IDE결정

Database/ WAS의 결정

데모 프로젝트의 작성 및 실행

# 개발을 위한 준비 (JDK1.8)

- JDK1.8사용
- 환경 변수 설정도 같이 진행할 것

## Java SE 8u171/ 8u172

Java SE 8u171 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u172 is a patch-set update, including all of 8u171 plus additional bug fixes (described in the release notes).

[Learn more](#) ▶

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
  - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
  - [JDK ReadMe](#)
  - [JRE ReadMe](#)

JDK

DOWNLOAD ↕

### 사용자 변수 편집

변수 이름(N):

JAVA\_HOME

변수 값(V):

C:\Program Files\Java\jdk1.8.0\_171\

디렉터리 찾아보기(D)...

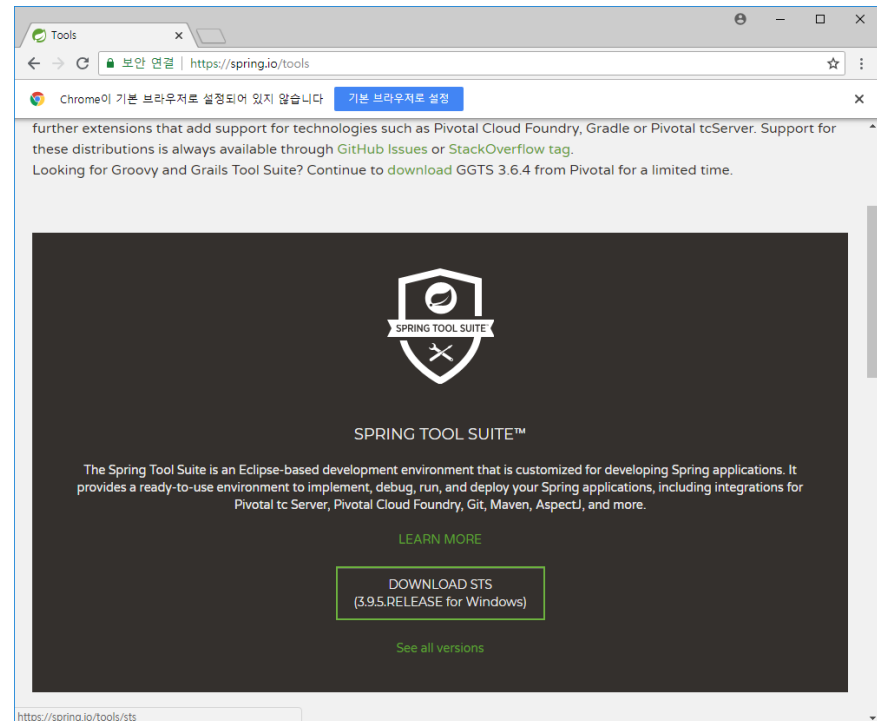
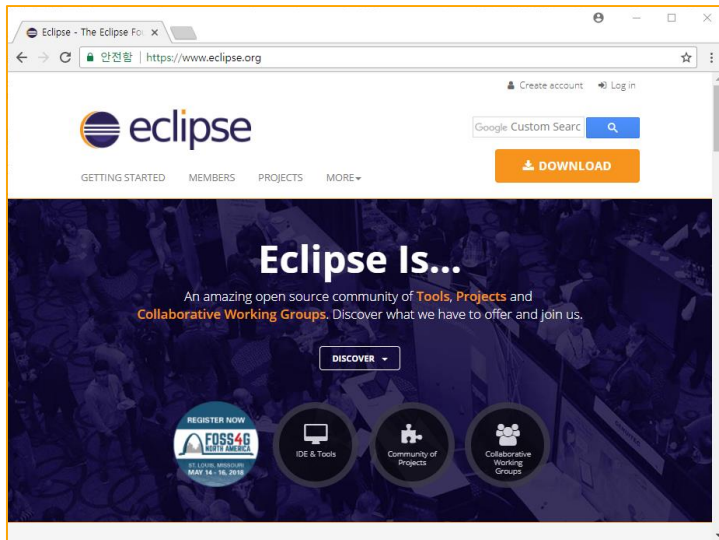
파일 찾아보기(F)...

확인

취소

# 개발을 위한 준비 (Eclipse or STS)

- 개발 도구 – Eclipse, STS(Spring Tool Suite), IntelliJ Ultimate 등
- Eclipse의 경우
  - STS 플러그인 추가 설치 후 사용
- STS의 경우
  - Eclipse와 별도로 다운로드 및 압축 해제



# STS의 플러그인

- <https://spring.io/tools3/sts/all>
- Eclipse 메뉴에서 'Help -> Install New Software'

## Update Site Archives

You can download archived versions of the update sites, if you want to install STS into an existing Eclipse installation or if you want to update an existing STS installation without accessing the hosted version of the update repository.

ECLIPSE	ARCHIVE	SIZE
4.8.0	<a href="#">springsource-tool-suite-3.9.5.RELEASE-e4.8.0-updatesite.zip</a>	163MB
4.7.3a	<a href="#">springsource-tool-suite-3.9.5.RELEASE-e4.7.3a-updatesite.zip</a>	149MB

## Update Sites

If you want to install STS in an existing Eclipse installation, you can use one of the following update sites. Please choose the one that matches the Eclipse version you're running:

ECLIPSE	UPDATE SITES
4.8	<a href="http://download.springsource.com/release/TOOLS/update/e4.8/">http://download.springsource.com/release/TOOLS/update/e4.8/</a>
4.7	<a href="http://download.springsource.com/release/TOOLS/update/e4.7/">http://download.springsource.com/release/TOOLS/update/e4.7/</a>
4.6	<a href="http://download.springsource.com/release/TOOLS/update/e4.6/">http://download.springsource.com/release/TOOLS/update/e4.6/</a>

# Tomcat 다운로드

- JDK버전에 맞게 다운로드 및 압축해제

## Apache Tomcat Versions

Apache Tomcat® is an open source software implementation of the Java Servlet and JavaServer Pages technologies. Different versions of Apache Tomcat are available for different versions of the Servlet and JSP specifications. The mapping between [the specifications](#) and the respective Apache Tomcat versions is:

Servlet Spec	JSP Spec	EL Spec	WebSocket Spec	JASPIC Spec	Apache Tomcat Version	Latest Released Version	Supported Java Versions
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.10	8 and later
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.32	7 and later
3.1	2.3	3.0	1.1	N/A	8.0.x (superseded)	8.0.53 (superseded)	7 and later
3.0	2.2	2.2	1.1	N/A	7.0.x	7.0.90	6 and later (7 and later for WebSocket)
2.5	2.1	2.1	N/A	N/A	6.0.x (archived)	6.0.53 (archived)	5 and later
2.4	2.0	N/A	N/A	N/A	5.5.x (archived)	5.5.36 (archived)	1.4 and later
2.3	1.2	N/A	N/A	N/A	4.1.x (archived)	4.1.40 (archived)	1.3 and later
2.2	1.1	N/A	N/A	N/A	3.3.x (archived)	3.3.2 (archived)	1.1 and later

# Making Spring Project

---

## 프로젝트의 생성 방식

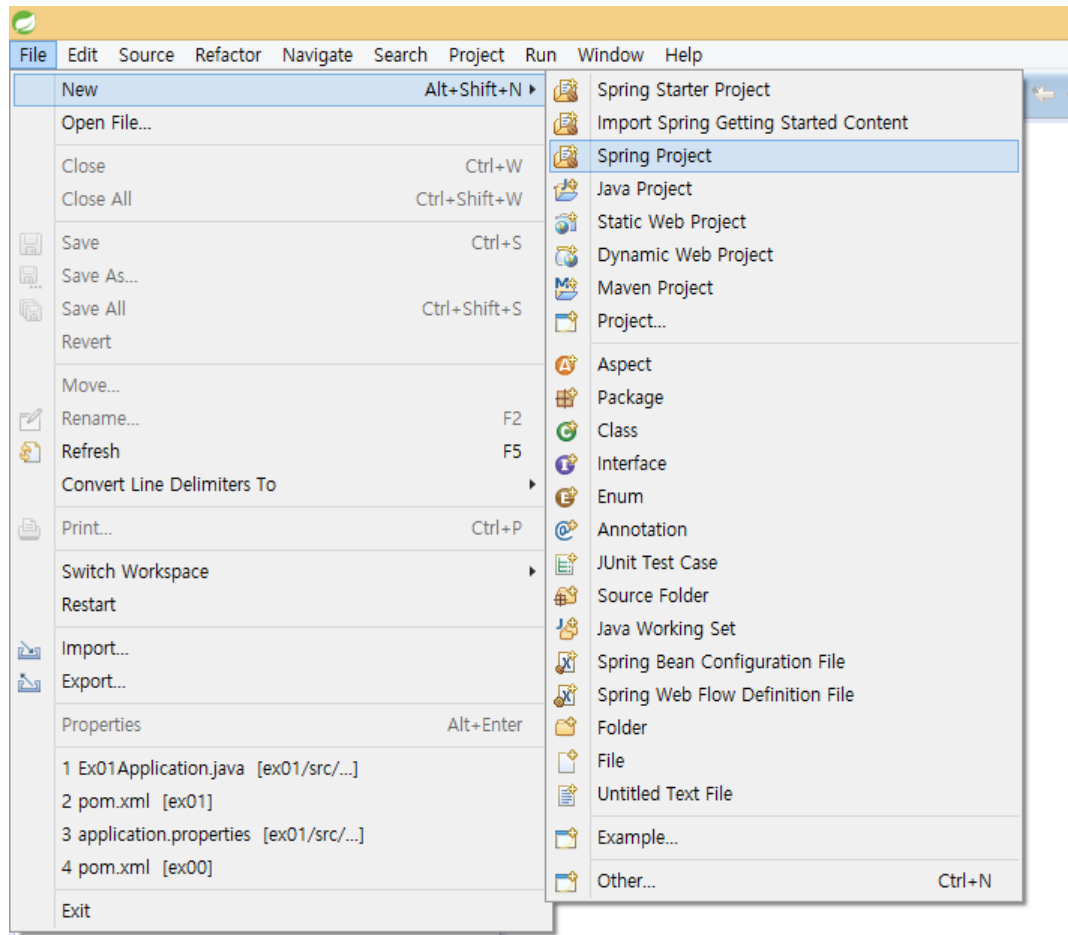
### Spring Starter Project

- Spring Boot 기반
- 책의 부록으로 설명

### Spring Project

- Spring MVC등의 개발 환경 제공
- 이 책의 예제로 사용

# Spring Project





# 스프링 프로젝트 생성

- Spring 프로젝트 중에서 'Spring Legacy Project' 생성

### Spring Legacy Project

Click 'Next' to load the template contents.

Project name:

☒ Use default location

Location:  [Browse...](#)

Select Spring version: [Default](#)

Templates:

- Simple Projects
  - Simple Java
  - Simple Spring Maven
  - Simple Spring Web Maven
- Batch
- GemFire
- Integration
- Persistence
- Simple Spring Utility Project
- Spring MVC Project**

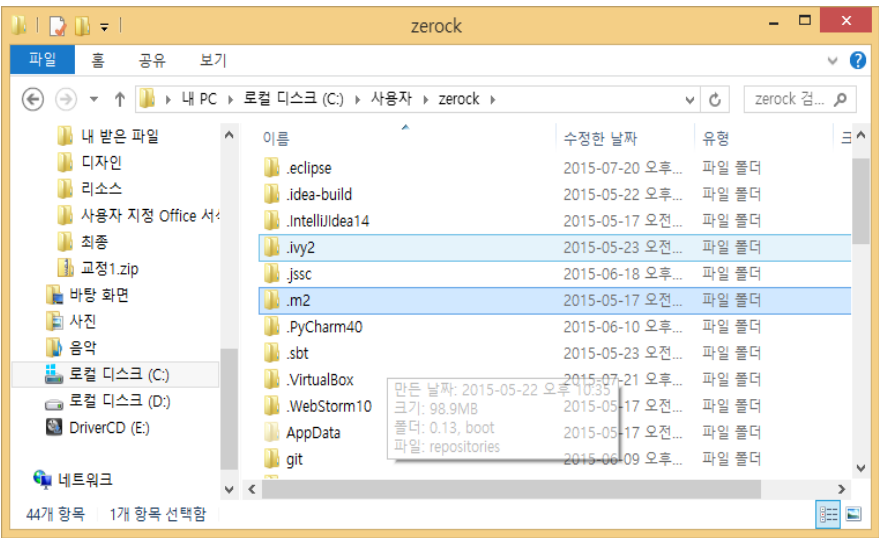
[requires downloading](#) [Configure templates...](#) [Refresh](#)

Description:  
A new Spring MVC web application development project

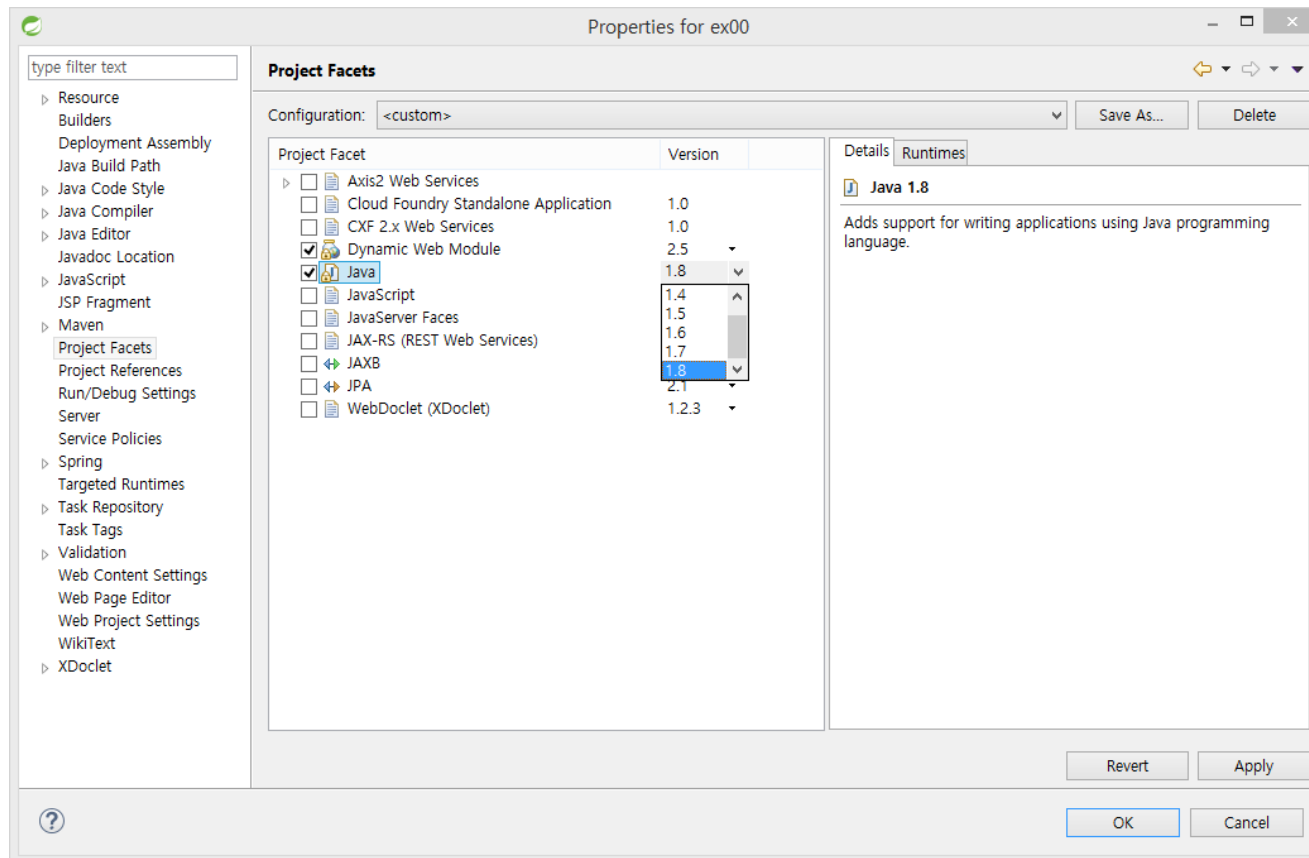
URL: <http://dist.springsource.com/release/STS/help/org.springframework.samples.mvc-3.2.2.zip>

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

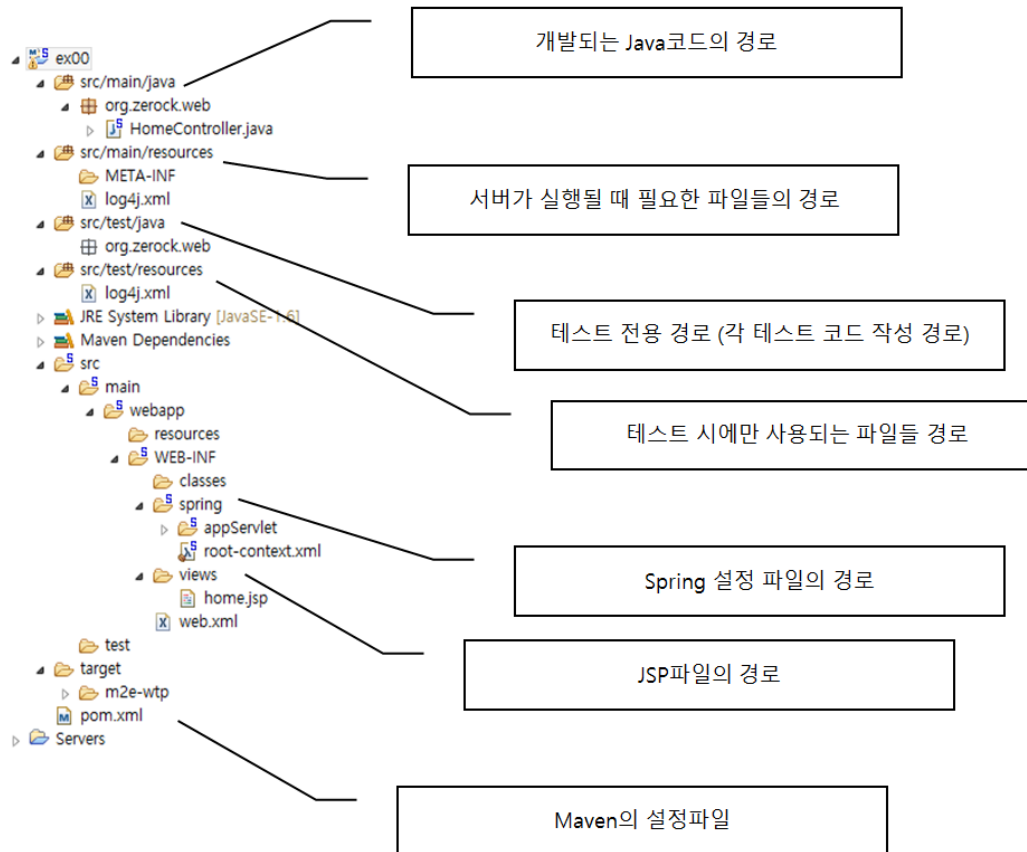
---



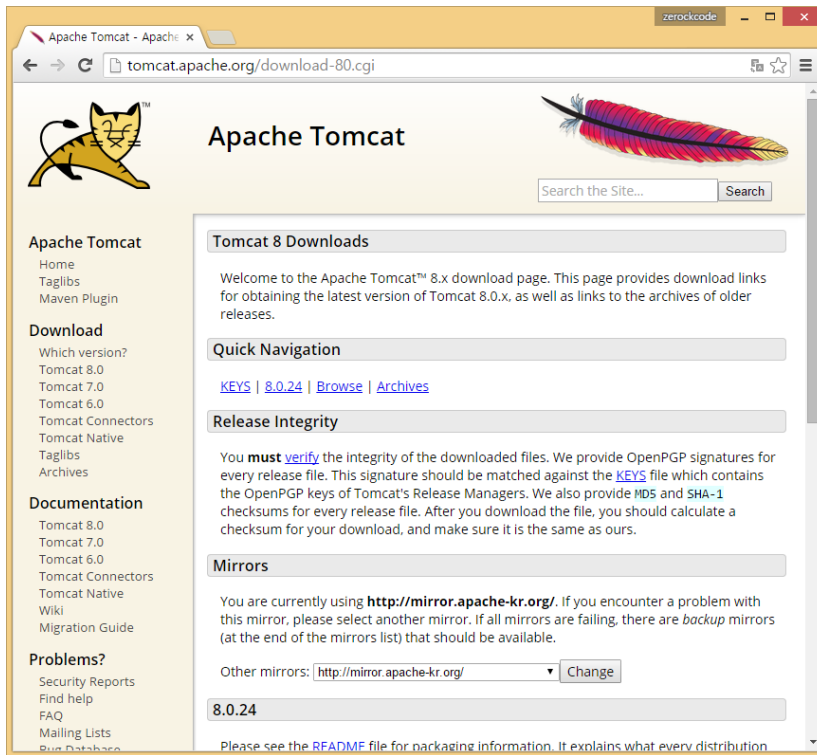
# JDK 버전 변경



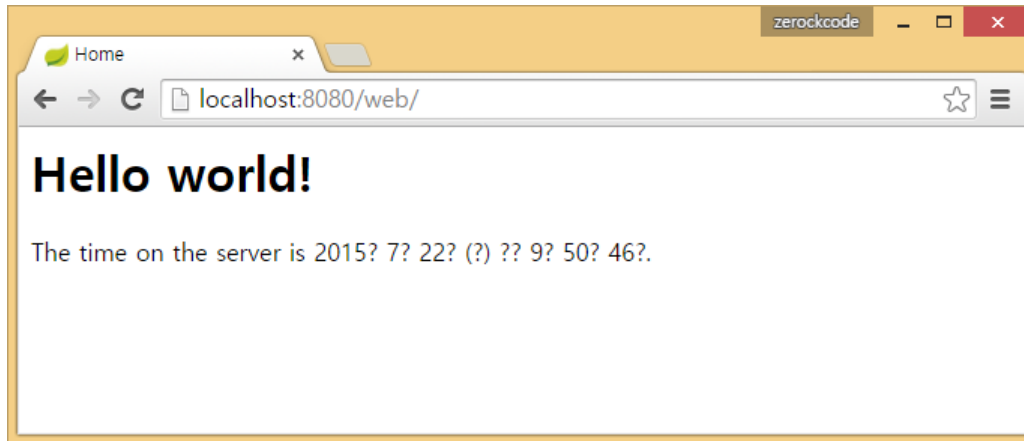
# Explain Project Folder



# Tomcat install(1) – Download



## 설정후 테스트 작업



# 스프링 프로젝트 생성

- Spring 프로젝트 중에서 'Spring Legacy Project' 생성

### Spring Legacy Project

Click 'Next' to load the template contents.

Project name:

☒ Use default location

Location:  [Browse...](#)

Select Spring version: [Default](#)

Templates:

- Simple Projects
  - Simple Java
  - Simple Spring Maven
  - Simple Spring Web Maven
- Batch
- GemFire
- Integration
- Persistence
- Simple Spring Utility Project
- Spring MVC Project**

[requires downloading](#) [Configure templates...](#) [Refresh](#)

Description:  
A new Spring MVC web application development project

URL: <http://dist.springsource.com/release/STS/help/org.springframework.samples.mvc-3.2.2.zip>

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

# pom.xml의 수정

---

- 스프링 프로젝트의 버전 변경 => 5.2.8
- Java 버전 변경 및 'Maven -> Update Project'

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.5.1</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
    <compilerArgument>-Xlint:all</compilerArgument>
    <showWarnings>true</showWarnings>
    <showDeprecation>true</showDeprecation>
  </configuration>
</plugin>
```



# 프로젝트의 실행 확인 및 경로 조정

- Tomcat을 이용한 프로젝트 실행 및 경로를 '/'로 조정
  - Tomcat의 'Modules'를 이용
  - 프로젝트의 'Web Settings'를 이용해서 조정

```
정보: Initializing Spring FrameworkServlet 'appServlet'
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'appSe
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resourc
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annota
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "{[/],methods=[GET]}" onto p
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebA
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebA
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/**] onto handler 'org.spri
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization completed in 832 ms
6월 14, 2018 11:34:34 오전 org.apache.coyote.AbstractProtocol start
정보: Starting ProtocolHandler ["http-nio-8080"]
6월 14, 2018 11:34:34 오전 org.apache.coyote.AbstractProtocol start
정보: Starting ProtocolHandler ["ajp-nio-8009"]
6월 14, 2018 11:34:34 오전 org.apache.catalina.startup.Catalina start
정보: Server startup in 6577 ms
INFO : org.zerock.controller.HomeController - Welcome home! The client locale is ko_KR.
```

# 개발을 위한 준비 (Lombok 라이브러리)

- 컴파일시에 getter/setter, 생성자, toString( )등을 자동으로 생성해 주기 때문에 편리함
- <https://projectlombok.org/>
- 다운로드 후에 Eclipse에 추가 설치 필요



# Java 설정을 이용하는 경우

---

- 기존의 XML파일들을 제거하고 pom.xml에 아래의 설정을 추가한 후에 프로젝트를 업그레이드

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.2.0</version>
  <configuration>
    <failOnMissingWebXml>>false</failOnMissingWebXml>
  </configuration>
</plugin>
```

# Java 설정을 이용하는 경우의 설정

---

- root-context.xml => RootConfig클래스로 처리
- web.xml => WebConfig클래스로 처리
- @Configuration 을 이용하는 설정

# RootConfig클래스

---

RootConfig 클래스

```
package org.zerock.config;
```

```
import org.springframework.context.annotation.Configuration;
```

```
@Configuration
```

```
public class RootConfig {
```

```
}
```

# WebConfig클래스

---

WebConfig 클래스

```
package org.zerock.config;
```

```
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
```

```
public class WebConfig extends AbstractAnnotationConfigDispatcherServletInitializer {
```

```
    @Override
```

```
    protected Class<?>[] getRootConfigClasses() {
```

```
        // TODO Auto-generated method stub
```

```
        return new Class[] {RootConfig.class};;
```

```
    }
```

```
    @Override
```

```
    protected Class<?>[] getServletConfigClasses() {
```

```
        // TODO Auto-generated method stub
```

```
        return null;
```

```
    }
```

```
    @Override
```

```
    protected String[] getServletMappings() {
```

```
        // TODO Auto-generated method stub
```

```
        return null;
```

```
    }
```

```
}
```