

Spring MVC Setting

524730-1
2022년 봄학기
5/18/2022
박경신

Spring Framework

□ Spring Framework

- 웹 서버를 구현하기 위한 어플리케이션 프레임워크
- IoC (Inversion of Control) 제어의 역행
 - IoC는 메소드나 객체의 호출을 개발자가 결정하는 것이 아니라 외부에서 결정되는 것을 의미
 - Spring IoC 컨테이너는 ApplicationContext 또는 BeanFactory 중 하나를 사용, Bean을 만들고, 의존성을 역어주며 제공해주는 역할을 수행
- DI (Dependency Injection) 의존성 주입을 통한 객체 간의 관계
 - IoC가 일어날 때 스프링 내부에 있는 객체들(Bean) 간의 관계를 컨테이너가 자동적으로 연결해주는 것
- AOP (Aspect Oriented Programming) 관점 지향 프로그래밍
 - 시스템 보안, 로그, 트랙잭션 등 스프링에서는 이러한 '횡단 관심사'를 모듈로 분리해서 제작 가능. 공통 관심 사항을 프로그래밍하여 이 코드를 여러 코드에 짜 넣음(weave). AspectJ로 이미 사용되어 왔음.
- POJO (Plain Old Java Object) 기반 구조
 - 객체 간의 관계 구성 시 POJO (즉, Bean)의 구성만으로 가능
- MVC (Model-View-Controller) 구조

Spring MVC Setting

❑ Spring MVC Quick Start

■ **JDK 11.0.2 & Eclipse IDE 2021-12**

■ **Tomcat 9.0.58**

- ❑ Eclipse Server Setting: Eclipse -> Window -> Preferences -> Server -> Runtime Environments -> Tomcat9

■ **Common Logging 1.2**

- ❑ https://commons.apache.org/proper/commons-logging/download_logging.cgi
- ❑ Download commons-logging-1.2-bin.zip

■ **Maven 3.8.1 프로젝트 관리 도구 maven-war-plugin 3.3.1**

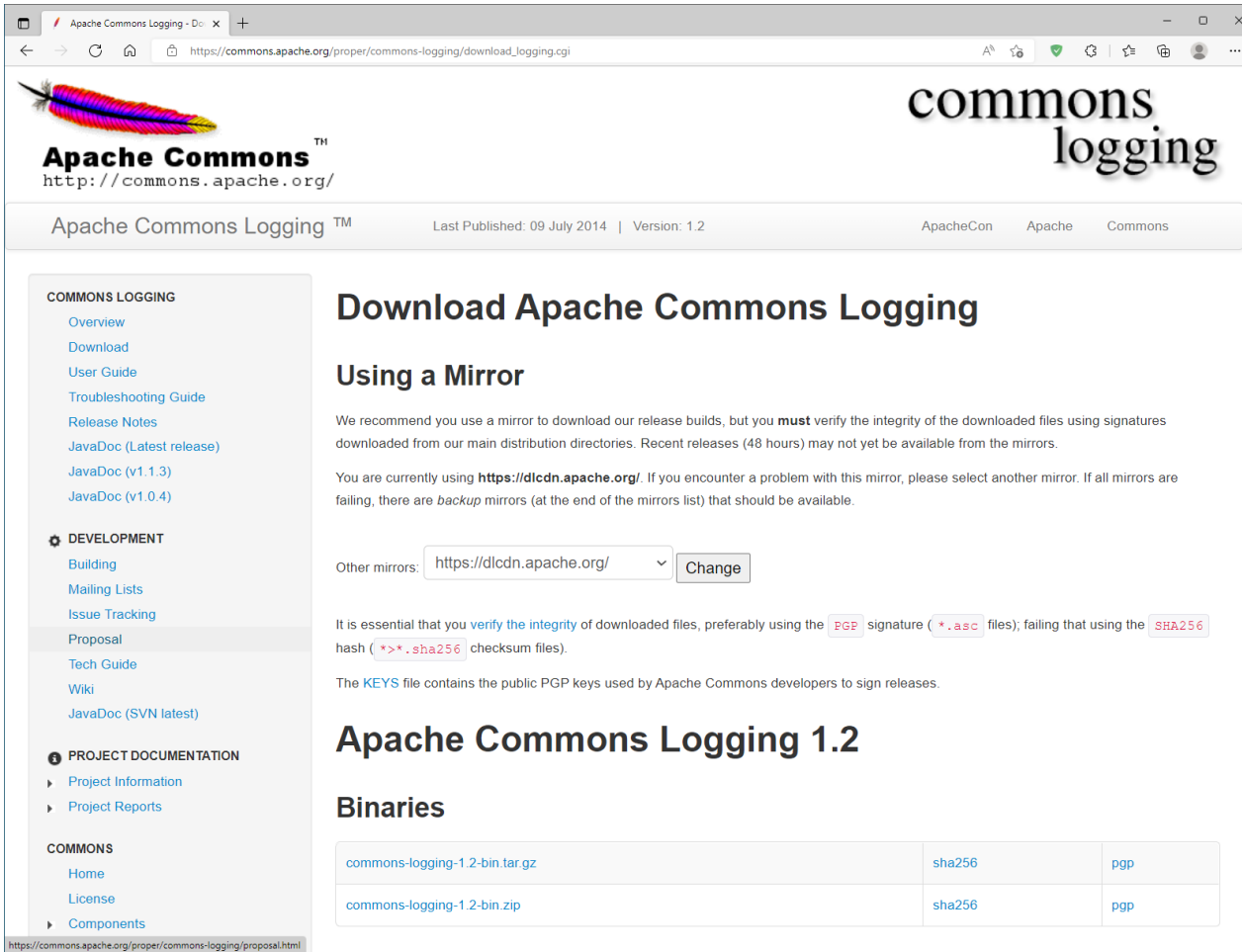
- ❑ <https://maven.apache.org/download.cgi>
- ❑ Download apache-maven-3.8.1-bin.zip

■ **Spring Tool Suite (STS) 5.2.8.RELEASE**

- ❑ <https://spring.io/tools/> Spring Tools for Eclipse
- ❑ Install Eclipse->Help->"Eclipse Marketplace" -> STS -> Spring Tools 3.9.14.RELEASE

Commons Logging

❑ https://commons.apache.org/proper/commons-logging/download_logging.cgi

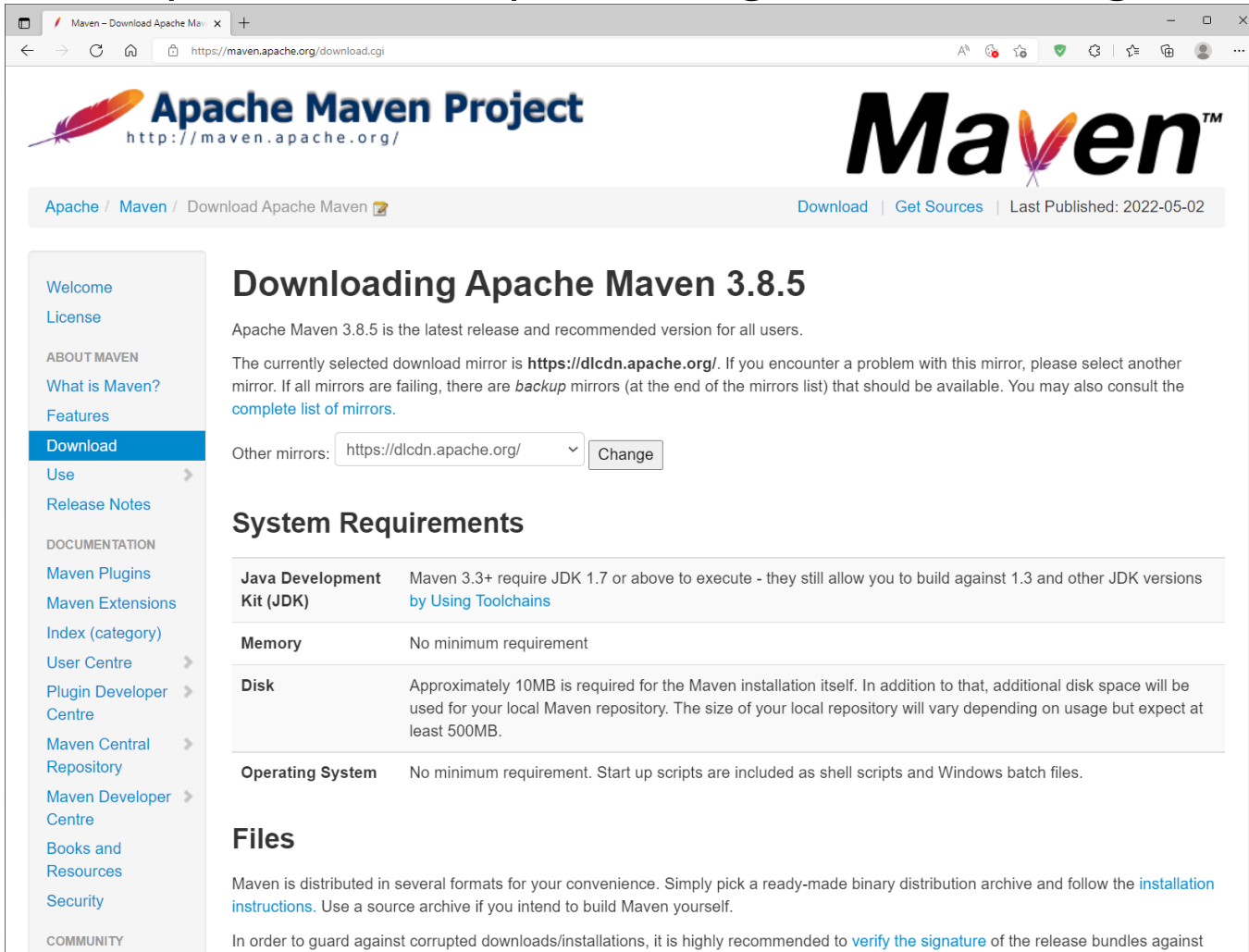


The screenshot shows the Apache Commons Logging download page. The browser address bar displays `https://commons.apache.org/proper/commons-logging/download_logging.cgi`. The page features the Apache Commons logo on the left and the 'commons logging' text on the right. A navigation bar includes 'Apache Commons Logging', 'Last Published: 09 July 2014', 'Version: 1.2', and links to 'ApacheCon', 'Apache', and 'Commons'. A left sidebar contains links for 'COMMONS LOGGING' (Overview, Download, User Guide, Troubleshooting Guide, Release Notes, JavaDoc (Latest release), JavaDoc (v1.1.3), JavaDoc (v1.0.4)), 'DEVELOPMENT' (Building, Mailing Lists, Issue Tracking, Proposal, Tech Guide, Wiki, JavaDoc (SVN latest)), 'PROJECT DOCUMENTATION' (Project Information, Project Reports), and 'COMMONS' (Home, License, Components). The main content area is titled 'Download Apache Commons Logging' and includes a section 'Using a Mirror' with a paragraph about mirrors and a form for selecting a mirror (currently set to `https://dldcn.apache.org/`). Below this, it states that users should verify the integrity of downloaded files using PGP signatures or SHA256 checksums. A section titled 'Apache Commons Logging 1.2' contains a 'Binaries' table.

Binary	SHA256	PGP
commons-logging-1.2-bin.tar.gz	sha256	pgp
commons-logging-1.2-bin.zip	sha256	pgp

Maven

❏ <https://maven.apache.org/download.cgi>



The screenshot shows the Apache Maven Project download page. The browser address bar displays `https://maven.apache.org/download.cgi`. The page header features the Apache Maven Project logo and the word "Maven" with a feather icon. A breadcrumb trail shows "Apache / Maven / Download Apache Maven". Navigation links include "Download", "Get Sources", and "Last Published: 2022-05-02".

The left sidebar contains a navigation menu with the following items:

- Welcome
- License
- ABOUT MAVEN
 - What is Maven?
 - Features
 - Download**
 - Use
 - Release Notes
- DOCUMENTATION
 - Maven Plugins
 - Maven Extensions
 - Index (category)
 - User Centre
 - Plugin Developer Centre
 - Maven Central Repository
 - Maven Developer Centre
 - Books and Resources
 - Security
- COMMUNITY

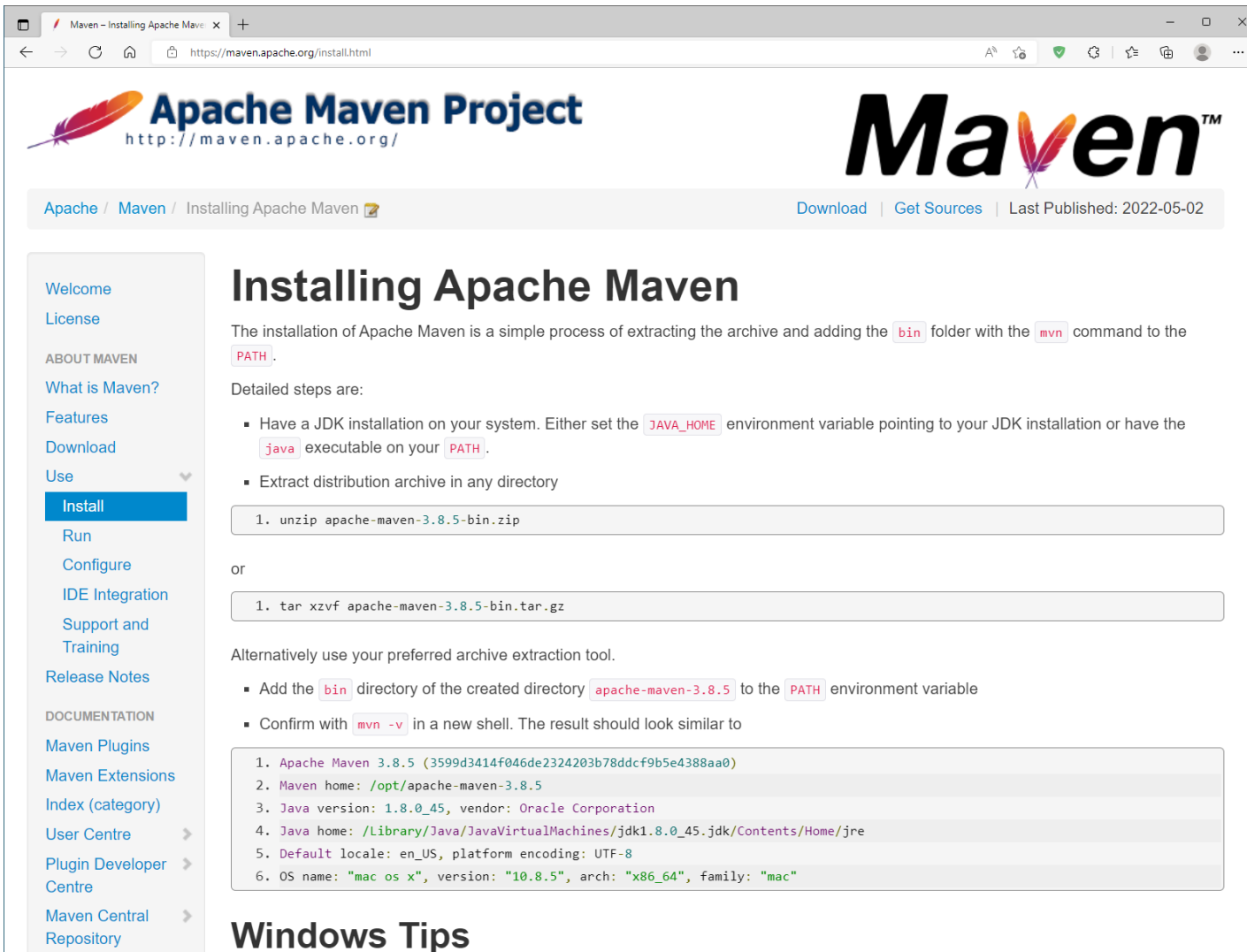
The main content area is titled "Downloading Apache Maven 3.8.5". It states that Apache Maven 3.8.5 is the latest release and recommended version for all users. The currently selected download mirror is `https://dlcdn.apache.org/`. A text input field shows "Other mirrors: `https://dlcdn.apache.org/`" with a "Change" button. Below this is the "System Requirements" section, which includes a table with the following information:

Requirement	Details
Java Development Kit (JDK)	Maven 3.3+ require JDK 1.7 or above to execute - they still allow you to build against 1.3 and other JDK versions by Using Toolchains
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Below the requirements is the "Files" section, which states that Maven is distributed in several formats for your convenience. It suggests picking a ready-made binary distribution archive and following the [installation instructions](#). It also mentions using a source archive if you intend to build Maven yourself. At the bottom, it recommends [verifying the signature](#) of the release bundles against the Apache signature key.

Maven

□ Maven



The screenshot shows the Apache Maven Project website. The browser address bar displays `https://maven.apache.org/install.html`. The page header includes the Apache Maven Project logo and the URL `http://maven.apache.org/`. The main heading is "Installing Apache Maven". The page content describes the installation process, mentioning the `bin` folder and the `mvn` command. It provides detailed steps for installing the JDK and extracting the Maven archive. The page also includes a sidebar with navigation links and a "Windows Tips" section at the bottom.

Apache / Maven / Installing Apache Maven  [Download](#) | [Get Sources](#) | Last Published: 2022-05-02

Installing Apache Maven

The installation of Apache Maven is a simple process of extracting the archive and adding the `bin` folder with the `mvn` command to the `PATH`.

Detailed steps are:

- Have a JDK installation on your system. Either set the `JAVA_HOME` environment variable pointing to your JDK installation or have the `java` executable on your `PATH`.
- Extract distribution archive in any directory

```
1. unzip apache-maven-3.8.5-bin.zip
```

or

```
1. tar xzvf apache-maven-3.8.5-bin.tar.gz
```

Alternatively use your preferred archive extraction tool.

- Add the `bin` directory of the created directory `apache-maven-3.8.5` to the `PATH` environment variable
- Confirm with `mvn -v` in a new shell. The result should look similar to

```
1. Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
2. Maven home: /opt/apache-maven-3.8.5
3. Java version: 1.8.0_45, vendor: Oracle Corporation
4. Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/jre
5. Default locale: en_US, platform encoding: UTF-8
6. OS name: "mac os x", version: "10.8.5", arch: "x86_64", family: "mac"
```

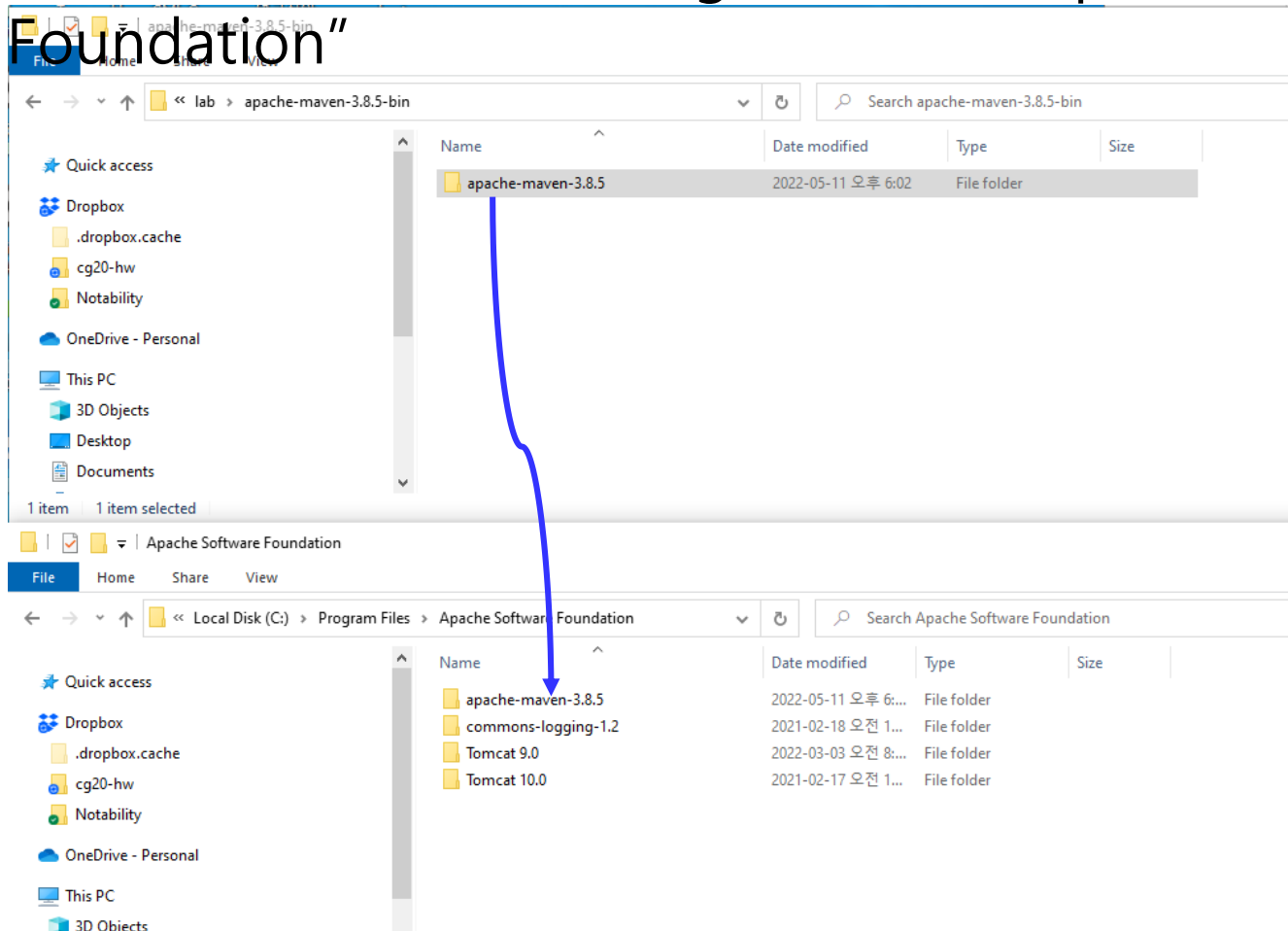
Windows Tips

Left Sidebar:

- Welcome
- License
- ABOUT MAVEN
- What is Maven?
- Features
- Download
- Use
- Install**
- Run
- Configure
- IDE Integration
- Support and Training
- Release Notes
- DOCUMENTATION
- Maven Plugins
- Maven Extensions
- Index (category)
- User Centre
- Plugin Developer Centre
- Maven Central Repository

Maven

- Unzip apache-maven-3.8.5-bin.zip & Copy "apache-maven-3.8.5" to "C:\Program Files\Apache Software Foundation"



Maven

▣ Set path environment variable

New System Variable

Variable name:

MAVEN_HOME

Variable value:

C:\Program Files\Apache Software Foundation\apache-maven-3.8.5

Browse Directory...

Browse File...

OK

Edit environment variable

C:\DevTools\Anaconda3

C:\DevTools\Anaconda3\Library\mingw-w64\bin

C:\DevTools\Anaconda3\Library\usr\bin

C:\DevTools\Anaconda3\Library\bin

C:\DevTools\Anaconda3\Scripts

C:\Windows\system32

C:\Windows

C:\Windows\System32\Wbem

C:\Windows\System32\WindowsPowerShell\v1.0\

C:\Windows\System32\OpenSSH\

C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common

%JAVA_HOME%\bin

C:\Users\park\AppData\Local\Android\Sdk\platform-tools

C:\Program Files\Git\cmd

C:\Program Files\dotnet\

C:\Program Files (x86)\dotnet\

%MAVEN_HOME%\bin

New

Edit

Browse...

Delete

Move Up

Move Down

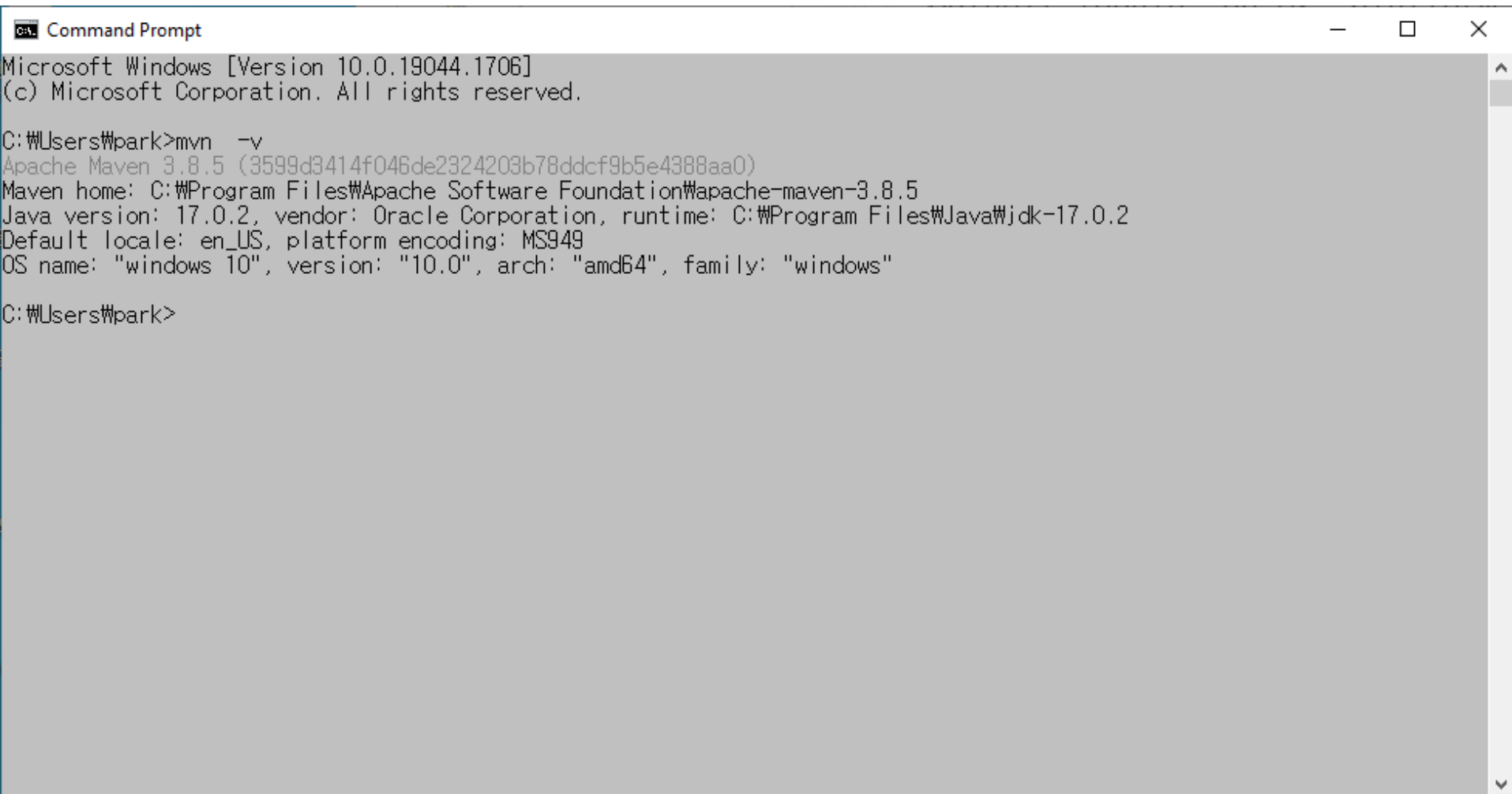
Edit text...

OK

Cancel

Maven

□ Run "mvn -v"



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\park>mvn -v
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: C:\Program Files\Apache Software Foundation\apache-maven-3.8.5
Java version: 17.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17.0.2
Default locale: en_US, platform encoding: MS949
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\park>
```

Spring Tool Suite (STS)

- ❑ Eclipse -> Help -> Eclipse Marketplace -> Find: STS -> Spring Tools 3 (Standalone Edition) 3.9.14.RELEASE -> Install

The screenshot shows the Eclipse IDE interface. On the left, the 'Help' menu is open, displaying various options. The 'Eclipse Marketplace...' option is highlighted. On the right, the 'Eclipse Marketplace' window is open, showing search results for 'STS'. The search results list two items: 'Spring Tools 3 (Standalone Edition) 3.9.14.RELEASE' and 'Spring Tools 3 Add-On for Spring Tools 4 3.9.22.RELEASE'. Both items have an 'Install' button next to them. The 'Spring Tools 3 (Standalone Edition)' item also shows a star rating of 100 and 277K installs. The 'Spring Tools 3 Add-On for Spring Tools 4' item shows a star rating of 323 and 520K installs. At the bottom of the marketplace window, there are buttons for '< Back', 'Install Now >', 'Finish', and 'Cancel'.

Help

- Welcome
- Help Contents
- Search
- Show Contextual Help
- Show Active Keybindings... Ctrl+Shift+L
- Tip of the Day
- Tips and Tricks...
- Report Bug or Enhancement...
- Cheat Sheets... Open the Eclipse Marketplace
- Eclipse User Storage >
- Perform Setup Tasks...
- Check for Updates
- Install New Software...
- Eclipse Marketplace...**
- About Eclipse IDE
- Donate
- Contribute

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation. Press the "more info" link to learn more about a solution.

Search Recent Popular Favorites Installed Giving IoT an Edge

Find: STS All Markets All Categories Go

Spring Tools 3 (Standalone Edition) 3.9.14.RELEASE

Spring Tools 3 (Standalone Edition) Attention - End of life: The Spring Tools 3 are no longer maintained and no longer updated for newer Eclipse versions. Please... [more info](#)

by VMware, EPL
[J2EE](#) [spring](#) [Spring IDE](#) [Cloud](#) [jee](#)

★ 100 Installs: 277K (8,674 last month) **Install**

Spring Tools 3 Add-On for Spring Tools 4 3.9.22.RELEASE

Spring Tools 3 Add-On for Spring Tools 4 End of life: This Spring Tools 3 Add-On for Spring Tools 4 is no longer maintained or updated. The final and last release... [more info](#)

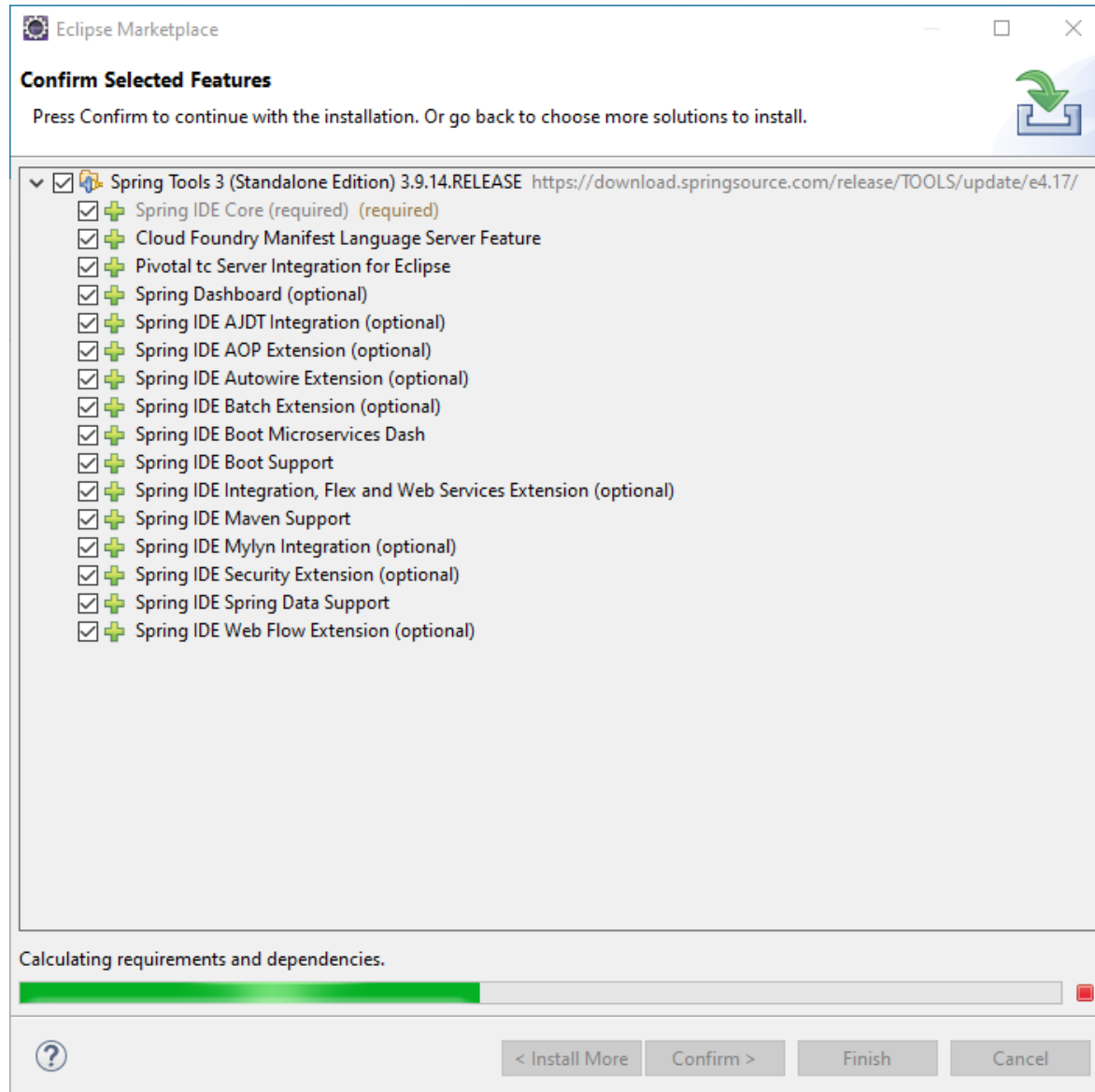
by VMware, EPL
[J2EE](#) [spring](#) [Spring IDE](#) [Cloud](#) [jee](#)

★ 323 Installs: 520K (7,822 last month) **Install**

Marketplaces

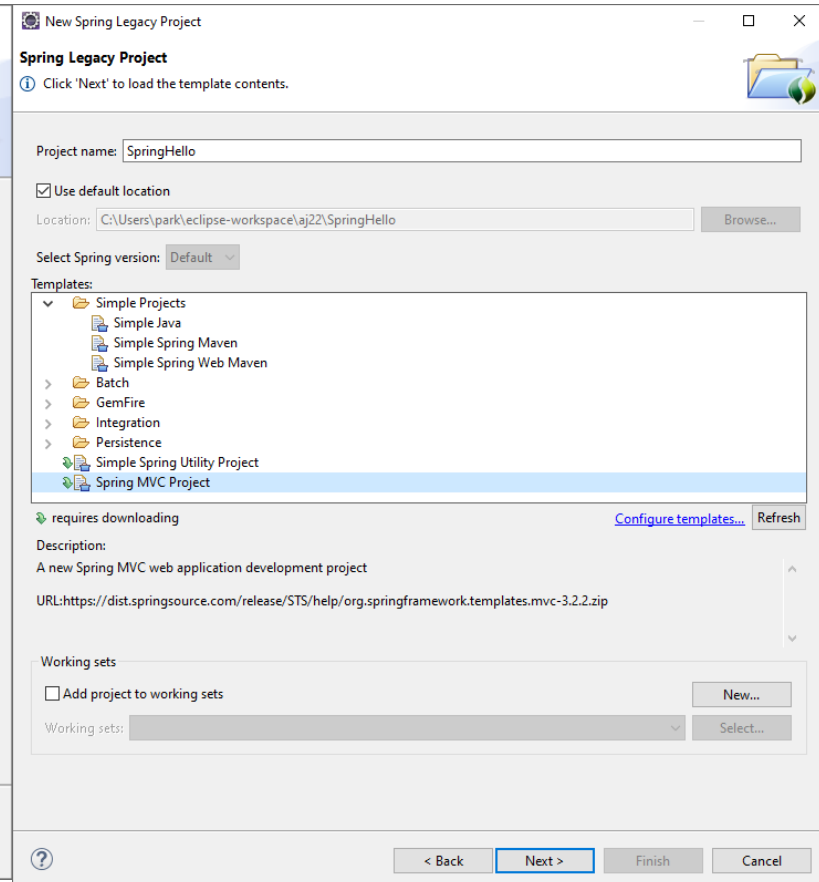
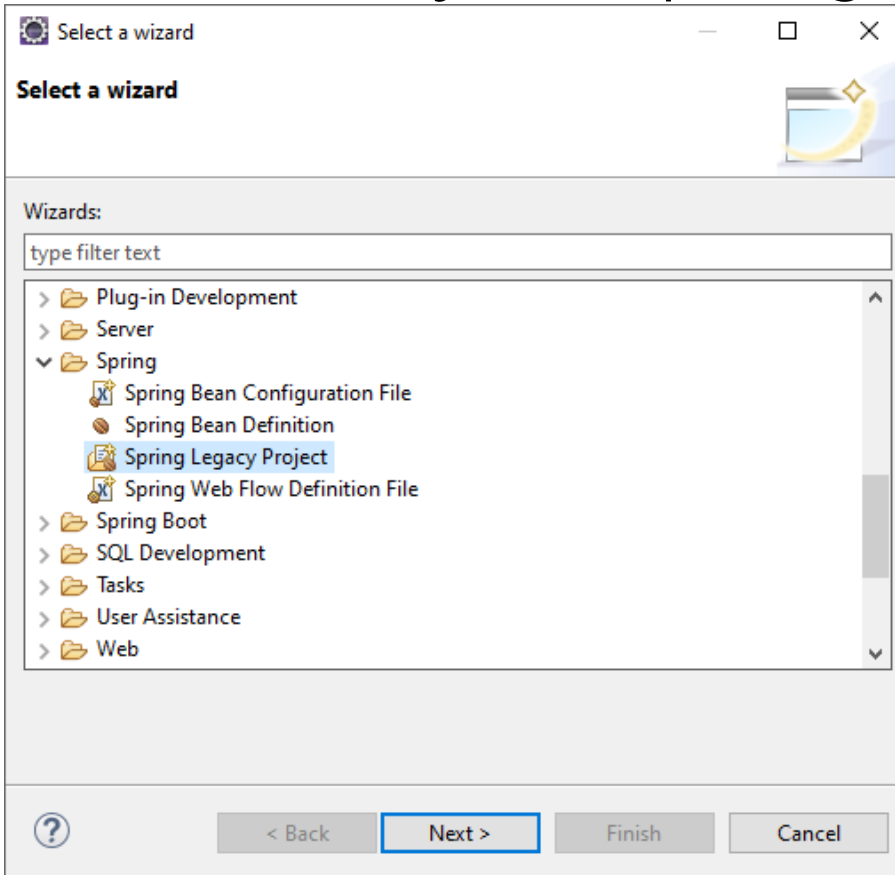
< Back Install Now > Finish Cancel

Spring Tool Suite (STS)

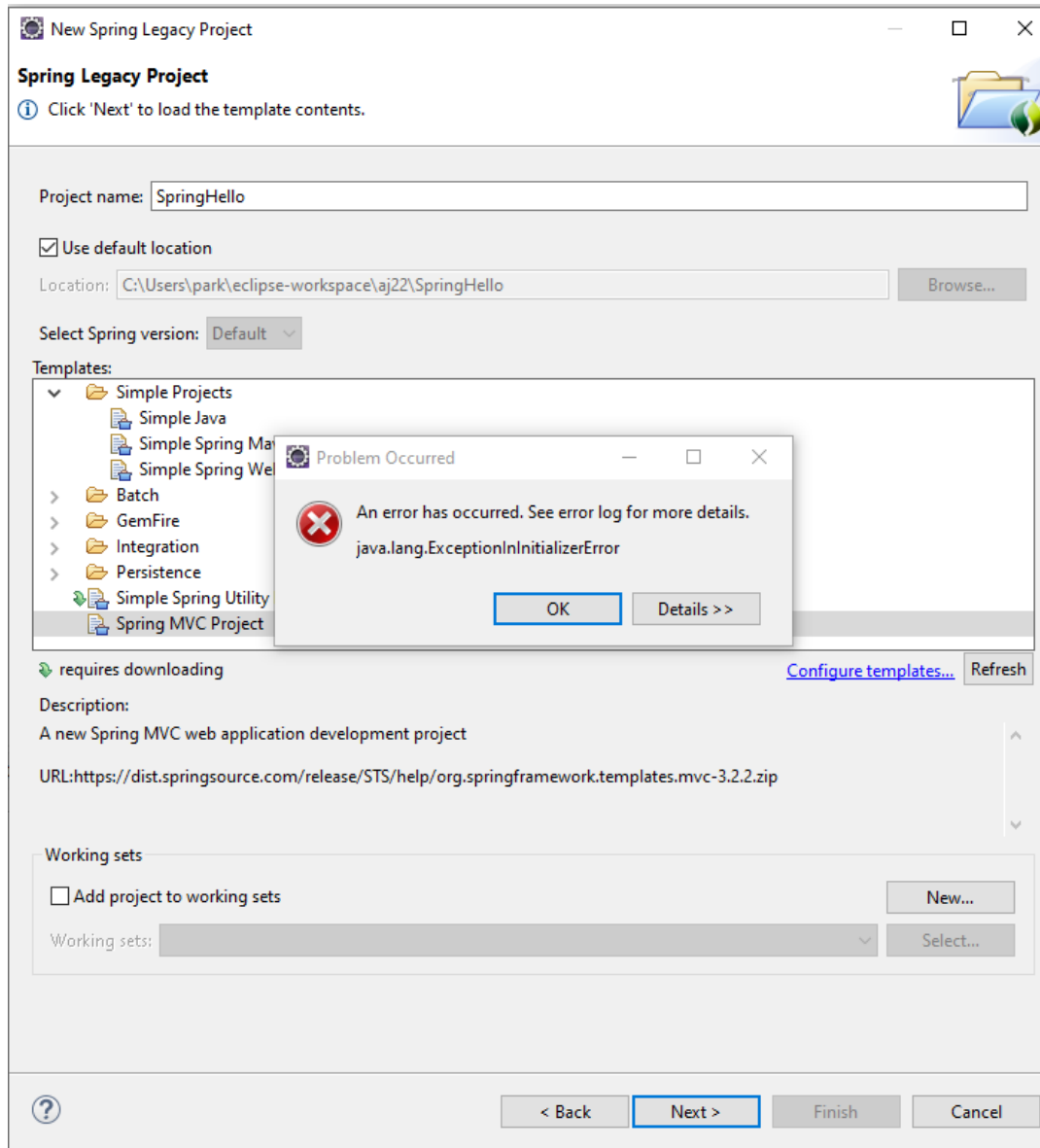


Eclipse Spring Legacy Project

- ❑ Eclipse -> File -> New -> Other -> Spring -> Spring Legacy Project -> Project name & Templates: Spring MVC Project -> package name -> Finish



java.lang.java.lang.ExceptionInInitializerError



Eclipse jee-2021-12 &
openjdk17 사용시
Spring Legacy Project 와
충돌문제
=> 자바 버전을
openjdk17에서 jdk11로
낮춰야 함

java.lang.java.lang.ExceptionInInitializerError



```
eclipse.ini - Notepad
C:\Users\park\eclipse\jee-2021-12\eclipse\eclipse.ini
File Edit Format View Help
-startup
plugins/org.eclipse.equinox.launcher_1.6.400.v20210924-0641.jar
--launcher.library
C:\Users\park\.p2\pool\plugins\org.eclipse.equinox.launcher.win32.win32.x86_64_1.2.400.v20211117-0650
-product
org.eclipse.epp.package.jee.product
-showsplash
C:\Users\park\.p2\pool\plugins\org.eclipse.epp.package.common_4.22.0.20211202-1200
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
C:\Program Files\Java\jdk-11.0.2\bin
-vmargs
-Dosgi.requiredJavaVersion=11
-Dosgi.instance.area.default=@user.home/eclipse-workspace
-Dsun.java.command=Eclipse
-XX:+UseG1GC
-XX:+UseStringDeduplication
--add-modules=ALL-SYSTEM
-Dosgi.requiredJavaVersion=11
-Dosgi.dataAreaRequiresExplicitInit=true
-Dorg.eclipse.swt.graphics.Resource.reportNonDisposed=true
-Xms256m
-Xmx2048m
--add-modules=ALL-SYSTEM
-Declipse.p2.max.threads=10
-Doomph.update.url=https://download.eclipse.org/oomph/updates/milestone/latest
-Doomph.redirection.index.redirection=index:/->http://git.eclipse.org/c/oomph/org.eclipse.oomph.git/plain/setups/
```

Eclipse Spring Legacy Project

- Spring Legacy Project -> Spring MVC Project -> package name -> Finish

New Spring Legacy Project

Project Settings - Spring MVC Project

Define project specific settings. Required settings are denoted by "**".

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

kr.ac.dankook.SpringHello

? < Back Next > Finish Cancel

Eclipse Spring Legacy Project

Properties for SpringHello

type filter text

- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Java Editor
- JSP Fragment
- Maven
- Project Facets**
- Project Natures
- Project References
- Run/Debug Settings
- Server
- Service Policies
- Spring
- Targeted Runtimes
- Task Repository
- Task Tags
- Validation
- Web Content Settings

Project Facets

Configuration: <custom>

Project Facet	Version
☐ Axis2 Web Services	
☐ CXF 2.x Web Services	1.0
☒ Dynamic Web Module	4.0
☒ Java	11
☐ JavaScript	1.0
☐ JavaServer Faces	2.3
☐ JAX-RS (REST Web Services)	1.1
☐ JAXB	2.2
☐ JPA	2.2
☐ WebDoclet (XDoclet)	1.2.3

Details | Runtimes

☒ Apache Tomcat v9.0

☐ Show all runtimes

Make Primary New...

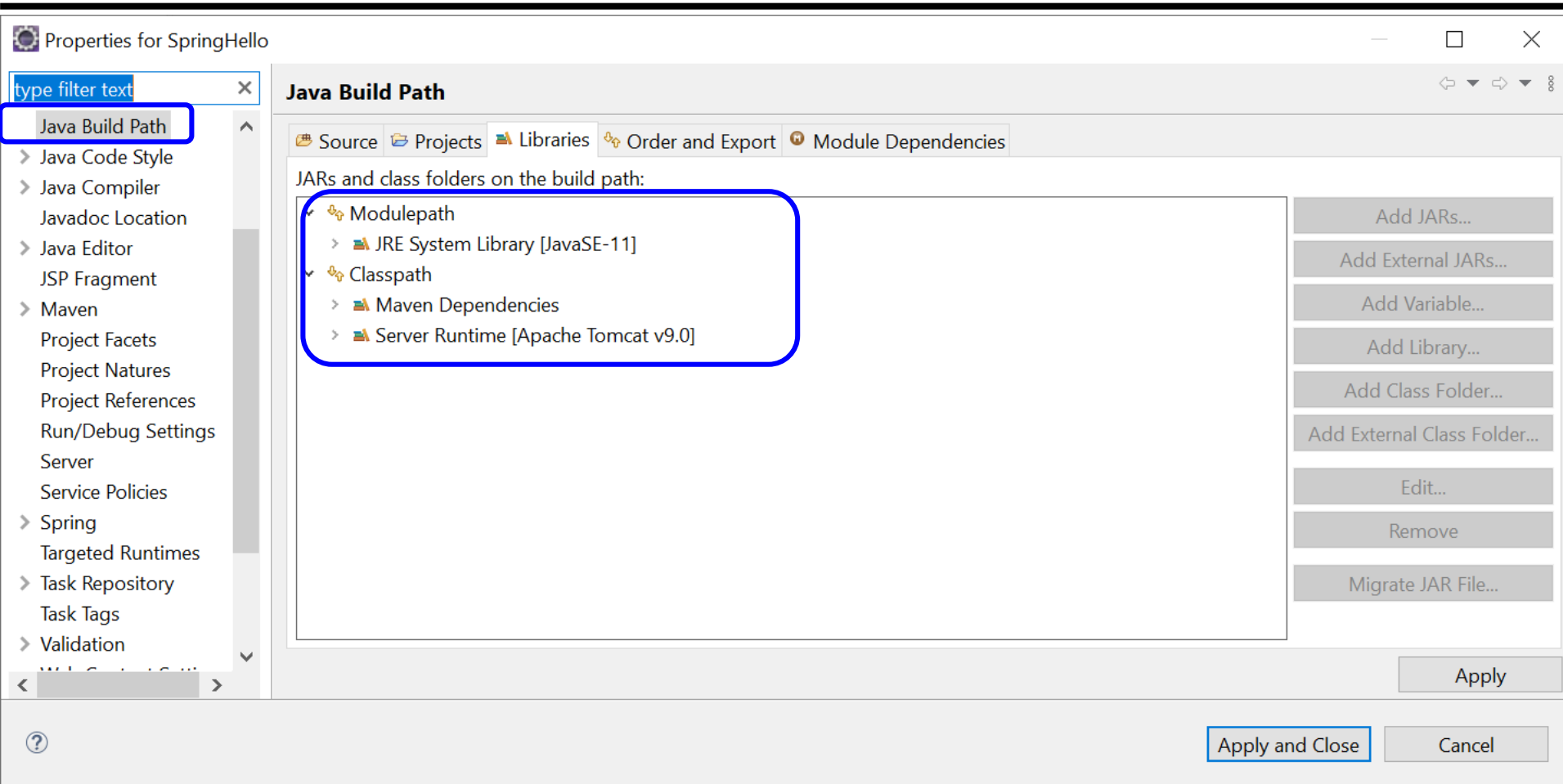
Runtime composition:

<no runtime selected>

Revert Apply

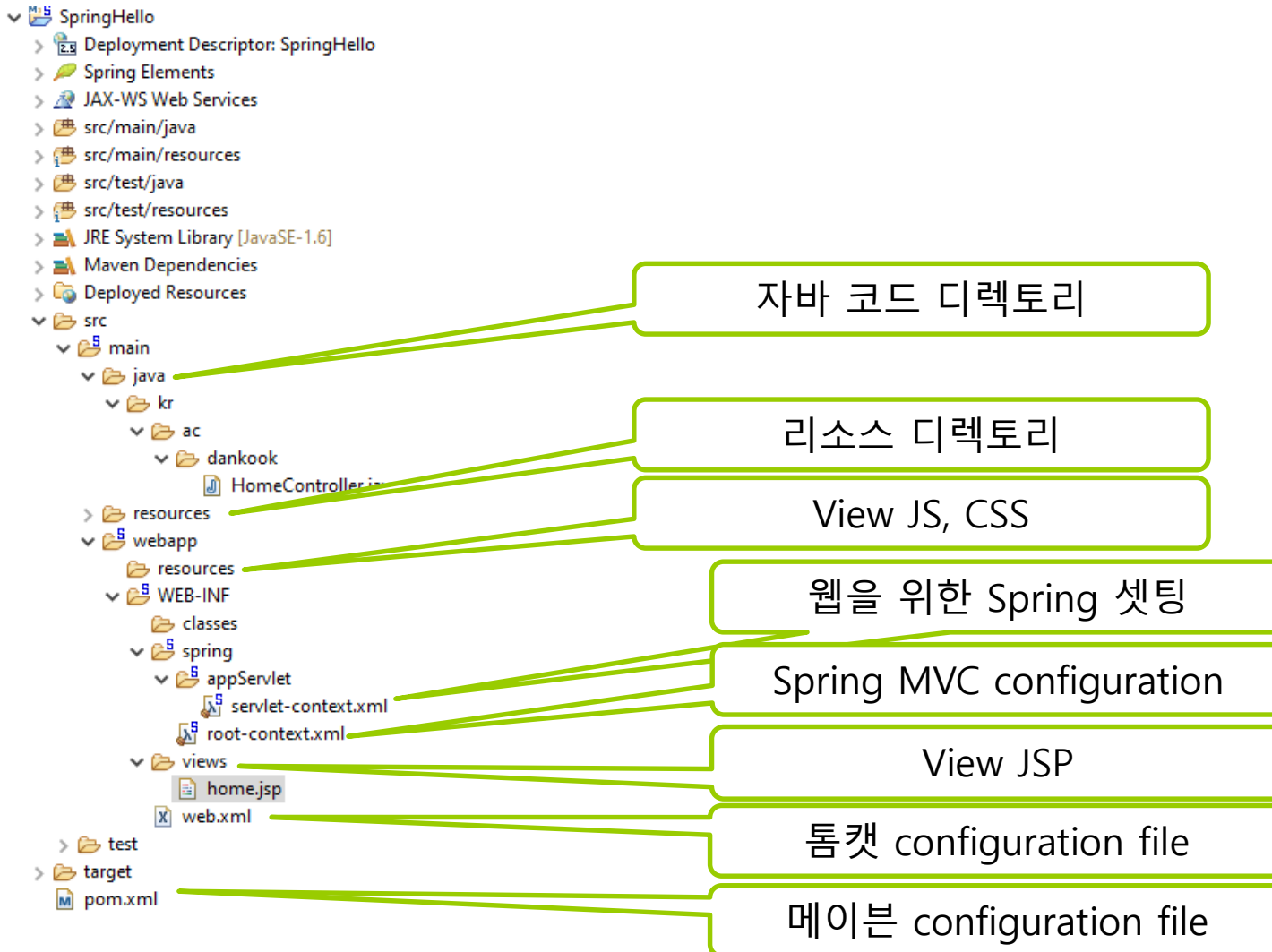
Apply and Close Cancel

Eclipse Spring Legacy Project



Spring MVC Project

□ Spring MVC Project 디렉토리 구조



pom.xml 수정

pom.xml

```
<properties>
<java-version>11</java-version>
<org.springframework-version>5.2.8.RELEASE</org.springframework-version>
<org.aspectj-version>1.6.10</org.aspectj-version>
<org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>11</source>
        <target>11</target>
        <compilerArgument>-Xlint:all</compilerArgument>
        <showWarnings>true</showWarnings>
        <showDeprecation>true</showDeprecation>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Eclipse Project -> Maven -> Update Project (Update Maven)

pom.xml 수정

pom.xml

```
<!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-war-plugin -->
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.3.1</version>
</plugin>
</plugins>
</build>
```

Eclipse Project -> Maven -> Update Project (Update Maven)

home.jsp 수정

home.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@ page session="false" %>
```

```
<html>
```

```
<head>
```

```
<title>Home</title>
```

```
</head>
```

```
<body>
```

```
<h1>
```

```
Hello world!
```

```
</h1>
```

```
<P> The time on the server is ${serverTime}. </P>
```

```
</body>
```

```
</html>
```

Eclipse Project -> Run As -> Maven Clean & Maven Install & Run on Server

Spring MVC Project 실행



Hello world!

The time on the server is 2022년 5월 13일 오전 10시 37분 46초 KST.

HomeController.java 수정

@Controller

HomeController.java

```
public class HomeController {  
    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);  
  
    @RequestMapping(value = "/", method = RequestMethod.GET)  
    public String home(Locale locale, Model model) {  
        logger.info("Welcome home! The client locale is {}. ", locale);  
        Date date = new Date();  
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG,  
            DateFormat.LONG, locale);  
        String formattedDate = dateFormat.format(date);  
        model.addAttribute("serverTime", formattedDate );  
        // add more attribute  
        String greetings = "Greetings, Spring MVC";  
        model.addAttribute("message", greetings);  
        return "home";  
    }  
}
```



serverTime과 message 속성이 모델에 추가됨

home.jsp 수정

home.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@ page session="false" %>
```

```
<html>
```

```
<head>
```

```
<title>Home</title>
```

```
</head>
```

```
<body>
```

```
<h1>
```

```
Hello world! ${message}
```

```
</h1>
```

```
<P> The time on the server is ${serverTime}. </P>
```

```
</body>
```

```
</html>
```

Eclipse Project -> Run As -> Maven Clean & Maven Install & Run on Server

Spring MVC Project 실행

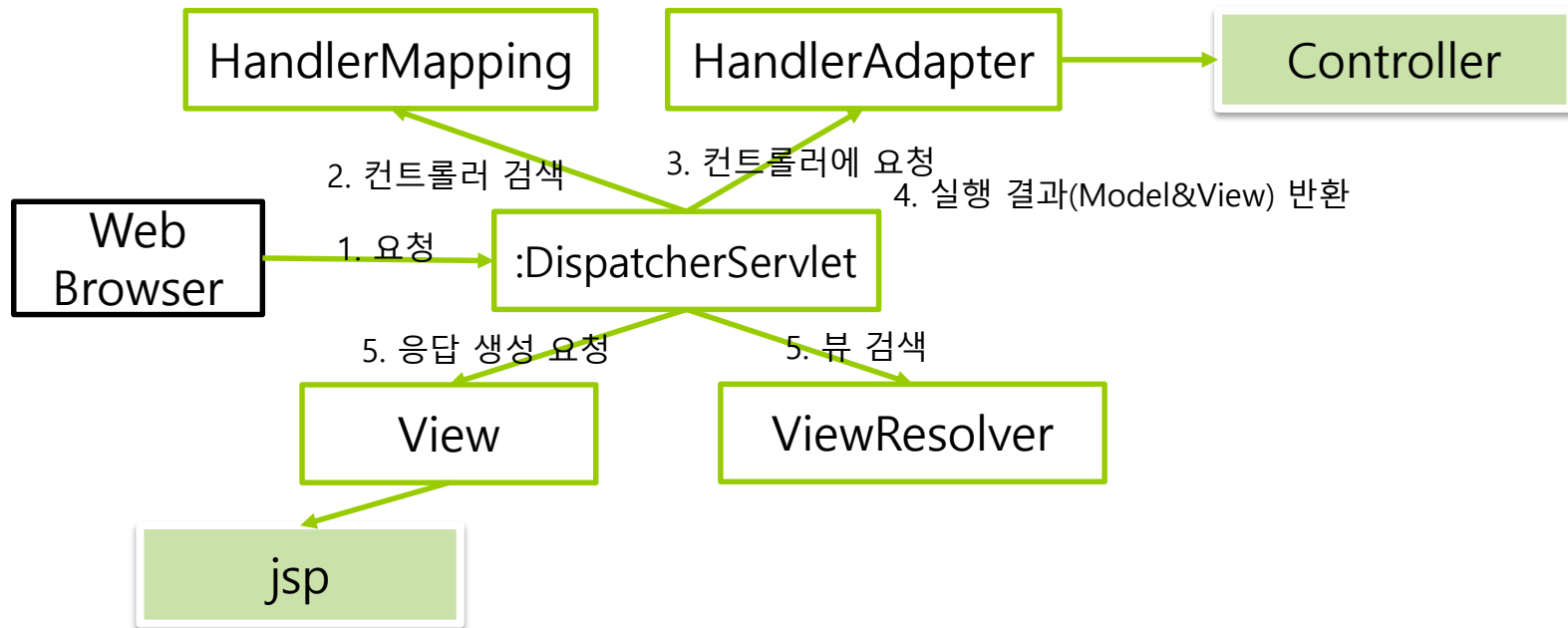


Hello world! Greetings, Spring MVC

The time on the server is 2022년 5월 13일 오전 10시 40분 14초 KST.

기본 흐름

- MVC 모델 특성상 누가 누구를 호출하는지 직접적으로 표현되지 않음
- 전체 구조를 이해할 필요가 있음



스프링 MVC 설정 과정

- web.xml에 DispatcherServlet 설정
 - 어떤 경로 접근에 어떤 기능(컨트롤러) 할당할 것인지
- web.xml 에 캐릭터 인코딩 필터 설정
- 스프링 MVC 설정
 - HandlerMapping, HandlerAdapter 설정
 - ViewResolver 설정

jsp 뷰를 생성하는데 사용

가장 심플하게는
<mvc:annotation-driven>태그를 xml
설정파일에 추가

java 설정 사용시
@EnableWebMvc

@Controller 태그
활성화

DispatcherServlet 설정

- 웹브라우저의 요청(<http://localhost:8080/hello>)
받아들여서 스프링 컨테이너를 생성
 - 컨테이너 설정 파일로는 /WEB-INF/서블릿이름-servlet.xml이 기본 설정됨
 - 설정 파일 목록을 만들기 위해서는 contextConfigLocation 매개변수를 web.xml에 지정

```
<!-- Processes application requests -->
<servlet>
  <servlet-name>appServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

web.xml

DispatcherServlet 설정

□ servlet-context.xml 확인

servlet-context.xml

```
<!-- DispatcherServlet Context: defines this servlet's request-processing
infrastructure -->
<!-- Enables the Spring MVC @Controller programming model -->
<annotation-driven />
<!-- Handles HTTP GET requests for /resources/** by efficiently serving up static
resources in the ${webappRoot}/resources directory -->
<resources mapping="/resources/**" location="/resources/" />
<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the
/WEB-INF/views directory -->
<beans:bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<beans:property name="prefix" value="/WEB-INF/views/" />
<beans:property name="suffix" value=".jsp" />
</beans:bean>

<context:component-scan base-package="kr.ac.dankook" />
```

캐릭터 인코딩 필터 설정

- 한글 문자열의 전송 패러미터(localhost/사용자/) 처리를 위한 인자
- web.xml의 webapp 태그 내에 다음 내용 추가

```
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

MVC 구성요소 설정 (servlet-context.xml)

- HandlerMapping, HandlerAdapter, ViewResolver 설정
- 내장 매퍼와 어댑터 사용은 다음으로 해결
 - @Controller 어노테이티드 클래스를 컨트롤로 사용할 수 있도록 함

```
<!-- Enables the Spring MVC @Controller programming model -->  
<annotation-driven />
```

servlet-context.xml

- 뷰 리졸버는 아래를 통해 구현
 - /web-inf/views/ 아래의 .jsp 파일들을 뷰로 사용

```
<!-- Resolves views selected for rendering by @Controllers to .jsp resources  
in the /WEB-INF/views directory -->  
<beans:bean  
class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
<beans:property name="prefix" value="/WEB-INF/views/" />  
<beans:property name="suffix" value=".jsp" />  
</beans:bean>
```

컨트롤러 경로 매핑(라우팅) (web.xml)

- 사용자가 요청하는 모든 경로에 대해 appServlet 사용
 - *.do 와 같이 확장자로 기능을 할당하는 경우
 - /users/* 와 같이 경로의 이름으로 기능을 할당하는 경우
 - 다음 코드는 전체 경로에 대해 **appServlet**을 할당하고 있음
 - jsp 요청을 제외한 나머지 모든 요청을 appServlet이 받아들이게 됨
 - 스프링 mvc 설정파일에 <mvc:default-servlet-handler />를 추가하여 톰캣의 404오류 페이지도 스프링이 핸들링하도록 할 수 있음

```
<servlet-mapping>  
  <servlet-name>appServlet</servlet-name>  
  <url-pattern>/</url-pattern>  
</servlet-mapping>
```


컨트롤러 구현

@Controller 어노테이션

HomeController.java

```
public class HomeController {  
    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
```

@RequestMapping(value = "/", method = RequestMethod.GET) 경로매핑

```
public String home(Locale locale, Model model) {  
    logger.info("Welcome home! The client locale is {}. ", locale);  
    Date date = new Date();  
    DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG,  
        DateFormat.LONG, locale);  
    String formattedDate = dateFormat.format(date);
```

```
    model.addAttribute("serverTime", formattedDate );  
    // add more attribute  
    String greetings = "Greetings, Spring MVC";  
    model.addAttribute("message", greetings);
```

모델에 데이터 추가
serverTime과 message
속성이 모델에 추가됨

```
    return "home"; 뷰 이름 리턴
```

```
}
```

```
}
```