



**Application Framework Development Guide**

---

## I. SKIAF 개발환경 가이드

Index of Contents

I.1 개요 . . . . .	3
I.2 특징 . . . . .	3
I.3 JDK 설치 . . . . .	3
I.4 Java IDE 설치 및 plugin 설정 . . . . .	5
I.4.1 STS (Spring Tool Suite) . . . . .	5
I.4.2 Lombok . . . . .	7
I.4.3 DTD (Document Type Definition) . . . . .	9
I.4.4 Code Template . . . . .	11
I.4.5 Indentation (들여쓰기) . . . . .	12
I.4.6 플러그인 설정 . . . . .	17
I.5. Nexus 설정 . . . . .	18
I.6. 개발환경 설정 . . . . .	19

# I. SKIAF 개발환경 가이드

## I.1 개요

### [ 목적 ]

본 문서는 프로젝트에서 SKIAF Framework를 활용하여 어플리케이션을 개발하는 절차 및 방법을 기술한다. 이들 개발환경 설정시 유의할 사항과 가이드라인 등을 다룬다. 개발환경 세팅 방법을 가이드 함으로써, 초기 개발의 생산성과 효율성을 향상하고, 개발된 결과의 통일성을 유지하여 향후 유지보수의 편의성, 높은 가독성을 제공하기 위함이다.

### [ 문서의 범위 ]

프로젝트에서 SKIAF Framework를 활용하여 개발할때 필요한 local 개발환경 설정 및 유의할 사항과 가이드라인을 제공한다.

### [ 대상 ]

프로젝트에서 SKIAF Framework를 활용하여 개발하는 개발자를 대상으로 한다.

## I.2 특징

SKIAF Framework는 SKI 계열의 Web 프로젝트 개발 업무를 효율적으로 수행하기 위한 Java 기반의 프레임워크 라이브러리이며, 다음의 특징을 가진다.

- Spring Boot 기반의 Java OpenSource를 활용한 Framework이다.
- 세션관리, 보안필터 등 시스템 환경에 대한 Framework 본연의 기본 기능과, BCM이라고 불리우는 업무공통 라이브러리를 포함하고 있다.
- PaaS 및 On-premise 환경 모두 대응 가능하도록 Redis를 이용한 세션관리 기능을 제공한다.
- Maven 프로젝트 디렉토리 구조를 따른다.
- Alopex UI 기술을 차용한 UI 표준을 포함하고 있다.

SKI 웹 프로젝트 개발시에 본 SKIAF Framework의 소스를 내려받아 Template처럼 이용하여 개발하도록 하며,

- SKI DevOps가이드를 참고하여 SVN (또는 PaaS 환경인 경우 ZCP-Git)을 통한 형상관리를 사용하는 것을 권장한다.
- SKI DevOps가이드는 에너지/화학지능화O.G 담당자에 문의하며,
- ZCP-Git 사용방법에 대한 가이드는 플랫폼O.G 담당자에게 문의한다.

※ BCM은 Business Common Module의 약자이며, 사용자/권한관리, 프로그램/코드관리, 메시지관리 등 업무 시스템을 관리하기 위한 관리 기능을 지칭한다.

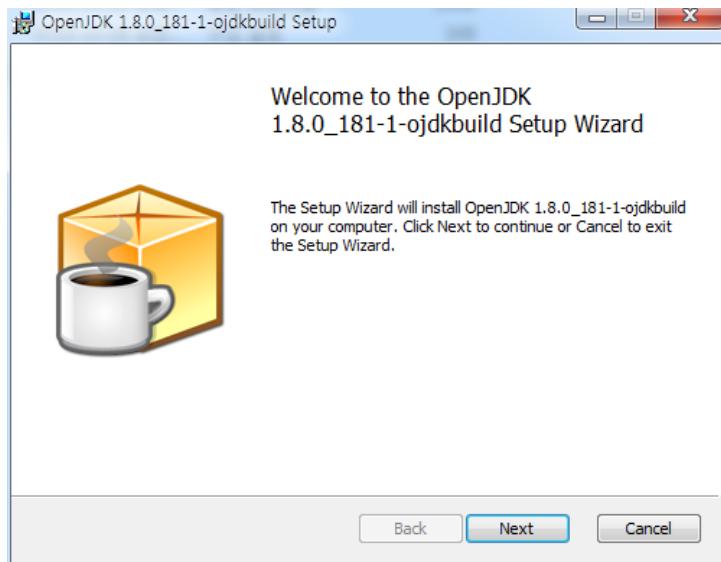
## I.3 JDK 설치

## [ JDK download ]

JDK는 2개 버전이 존재한다. 하나는 폐쇄적인 상업코드 기반의 Oracle JDK(유료)이고, 하나는 오픈소스기반의 OpenJDK이다.

둘간의 큰차이라면 글꼴라이브러리와 java web start 가 있다. 사용자의 웹브라우저에서 자바 애플릿을 실행하려면 필요한 기능이다. 서버 애플리케이션 개발자에게는 사용되지 않는 기능이다. 그래서 본 프로젝트는 OpenJDK 1.8 버전의 64bit를 다운받아서 준비한다.

## [ JDK 설치 ]



인터넷에서 파일을 다운받아 설치를 진행한다.

## [ JDK 설치확인 ]

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Win01876>java -version
openjdk version "1.8.0_181-1-ojdkbuild"
OpenJDK Runtime Environment (build 1.8.0_181-1-ojdkbuild-b13)
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)

C:\Users\Win01876>
```

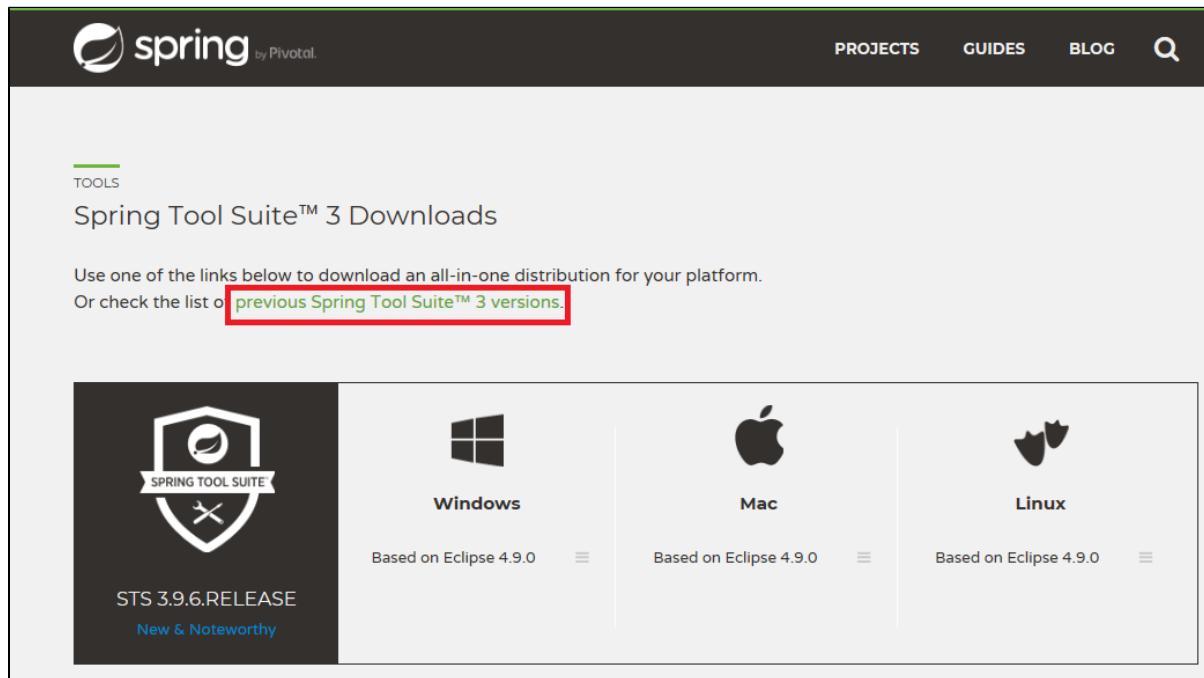
설치후에 java -version 명령에서 위와 같이 "open JDK"라는 글자가 나오면 정상 설치 된 상황이다.

## I.4 Java IDE 설치 및 plugin 설정

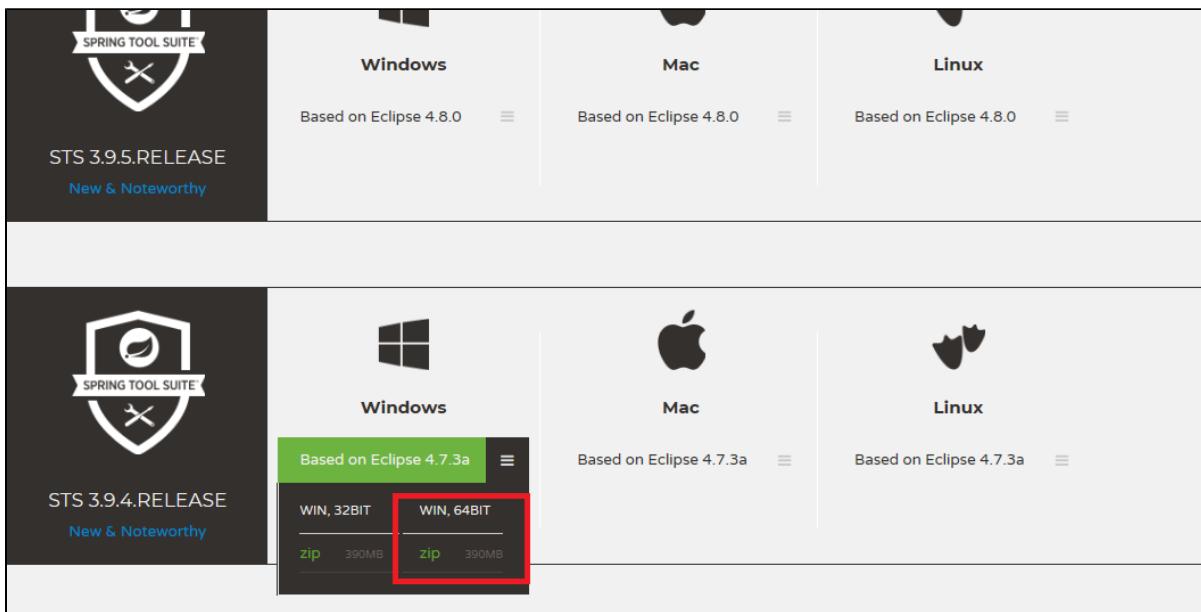
### I.4.1 STS (Spring Tool Suite)

[STS 다운로드]

- 다운로드 사이트 : <https://spring.io/tools/sts/all>



먼저 사이트로 들어가면 위와 같은 모습일 것이다. 자신의 PC 운영체제에 맞는 STS를 다운받으면 된다. STS 3.9.4 버전을 다운받아야 하므로 "previous Spring Tool Suit 3 versions"을 클릭한다.

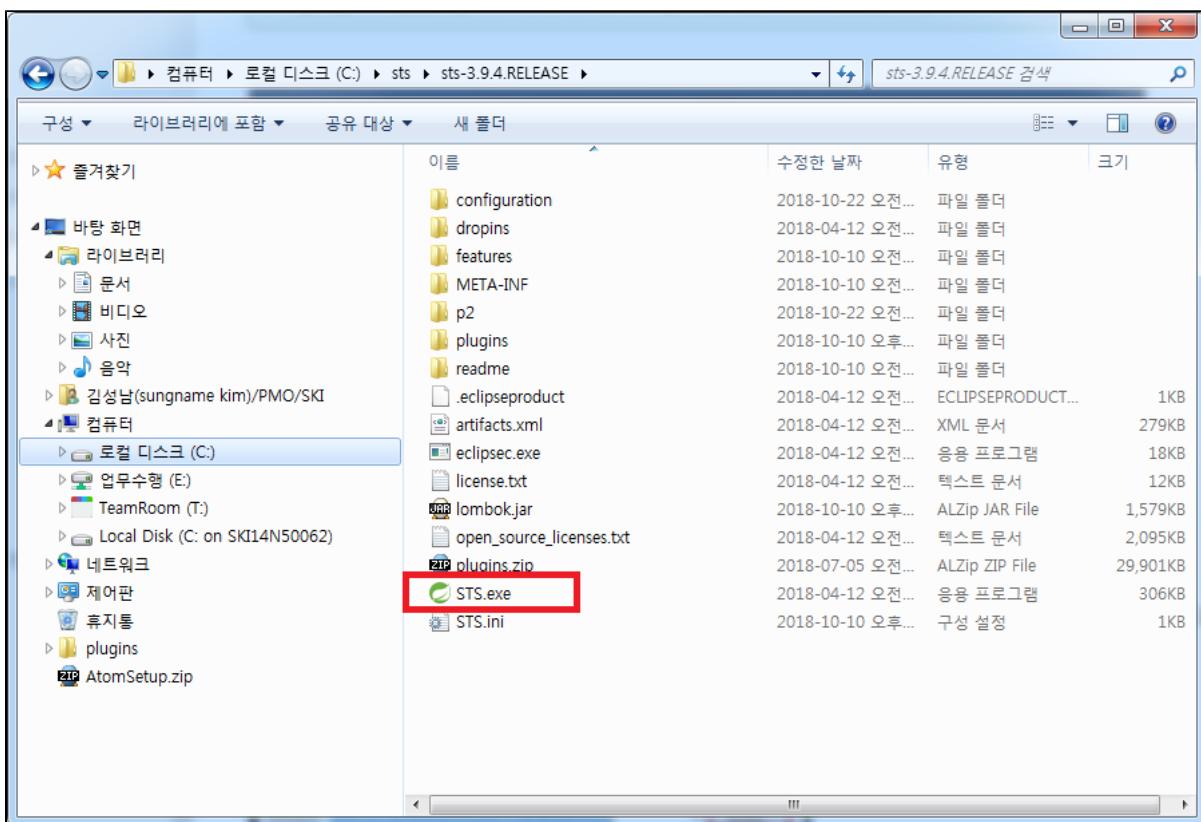


Window 아래 Base on Eclipse 4.7.3을 클릭한다. 여기에서 자신의 PC의 Bit에 맞는 STS를 다운로드한다.

### [STS 설치]

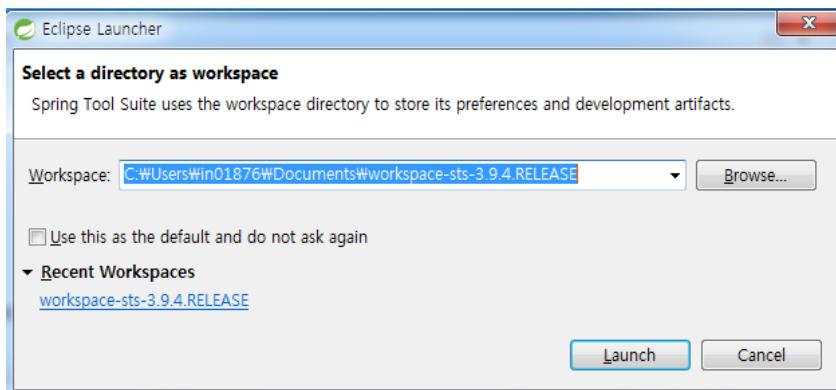
#### 도움말

압축풀기 추천경로는 "C:\₩" 이다. C:\₩ 에 바로 풀지 않으면 window 최대 path 길이 초과로 에러가 발생한다



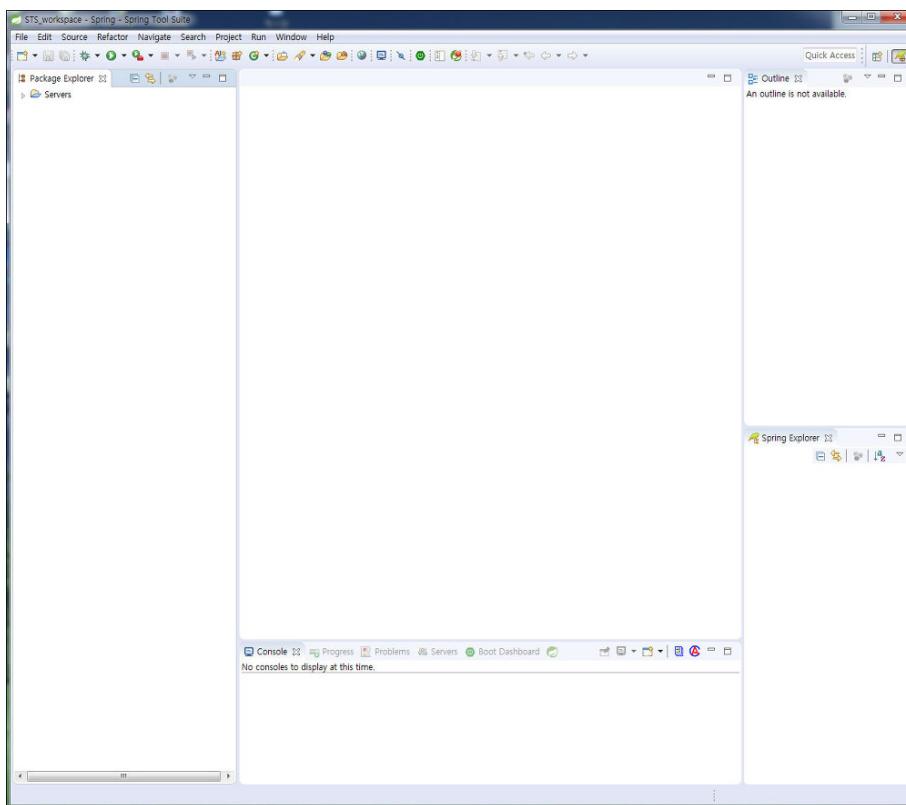
STS 파일을 다운받았다면 압축을 풀자. 압축 푼 파일에서 [sts-bundle] - [sts-3.9.4.RELEASE] 폴더에 들어가 보자. 그러면 STS.exe라는 파일이 있을

것이다. 이것이 STS의 실행파일이다.



파일을 실행하면 위와 같이 나타날 것이다. 여기서 Browser 버튼을 클릭하여 Workspace 폴더를 변경하자.

그리고 OK를 누른다.



Workspace를 지정 후 실행되면 위와 같이 나타날 것이다. 이렇게 나타나면 STS 설치가 끝이 난다.

## I.4.2 Lombok

[ Lombok 소개 ]

Lombok이란? 자바에서 DTO, VO, Domain Object 를 만들다보면, 멤버필드(프로퍼티)에 대한 Getter/Setter 메소드등 기계적으로 만드는 코드에 대해서 어노테이션을 통해서 코드 작성량과 코드의 길이를 줄여 반복되고 피곤한 코드를 줄여주는 것이 주된 목적인 자바 라이브러리이다

### [ Lombok 다운로드 ]

#### 1안. Lombok.jar 직접 다운로드

<http://projectlombok.org/download.html>에서 파일을 다운받아 설치한다.

#### 2안. 프로젝트의 pom.xml에 lombok 의존성 추가 (필요한 버전이 있다면 버전을 표시)



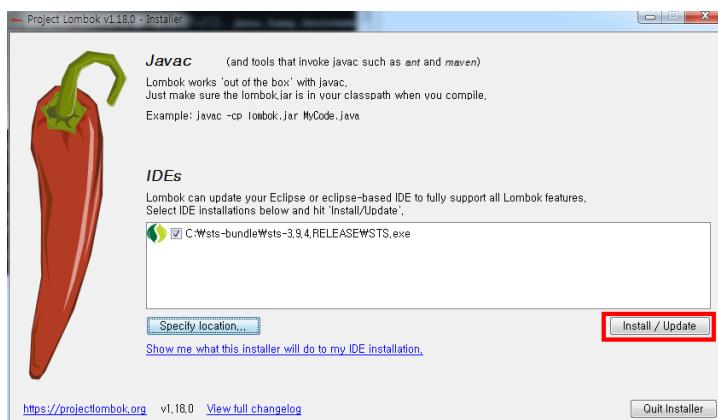
### [ Lombok 설치 ]

#### 1안. 콘솔창에서 "java -jar lombok.jar" 실행 (원도우에서는 더블클릭 가능)

- Lombok.jar 직접 다운로드 설치시에는 다운로드 폴더로 이동 후 실행한다.

#### 2안. Maven Dependency 추가 설치시에는 Maven Repository 폴더 이동 후 "java -jar lombok.jar" 실행

- 기본위치 : '~/.m2/repository/org/projectlombok/lombok/{해당버전}'

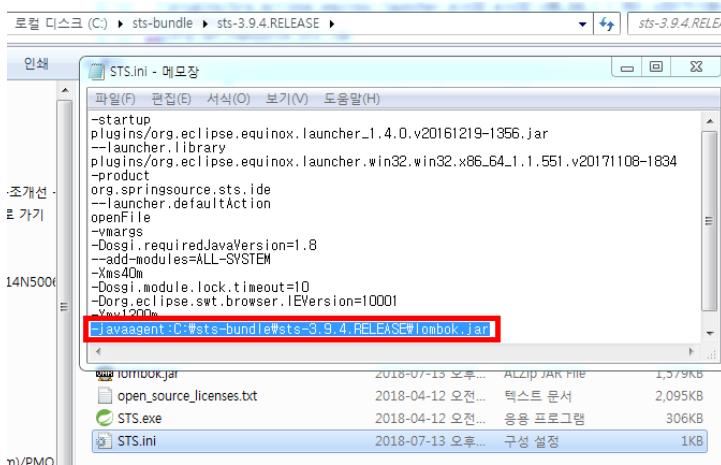


Install/Update 버튼을 이용하여 설치 한 후 Quite Installer 버튼을 눌러 종료한다.

(자동으로 설치되어 있는 목록이 나온다)

### [ Lombok 설치확인 ]

정상 설치시 STS.ini에 lombok agent가 설정되어있다.



### I.4.3 DTD (Document Type Definition)

#### [ DTD 소개 ]

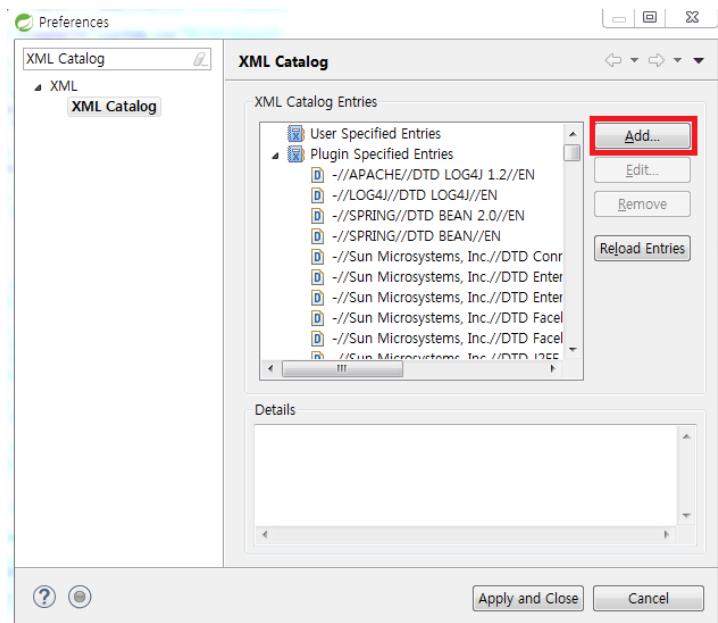
XML 규칙을 사용자가 직접 지정할 수 있도록 하는 것이다.

mybatis를 이용해 SQL쿼리를 작성할때 자동완성기능을 정상적으로 사용하기 위해 DTD 템플릿을 추가해야한다.

인터넷을 사용 할 수 없는 환경에서 XML 문서가 DTD를 참조하게 된다.

#### [ DTD 추가하기 ]

STS - Window - Preferences - XML - XML Catalog - Add 클릭



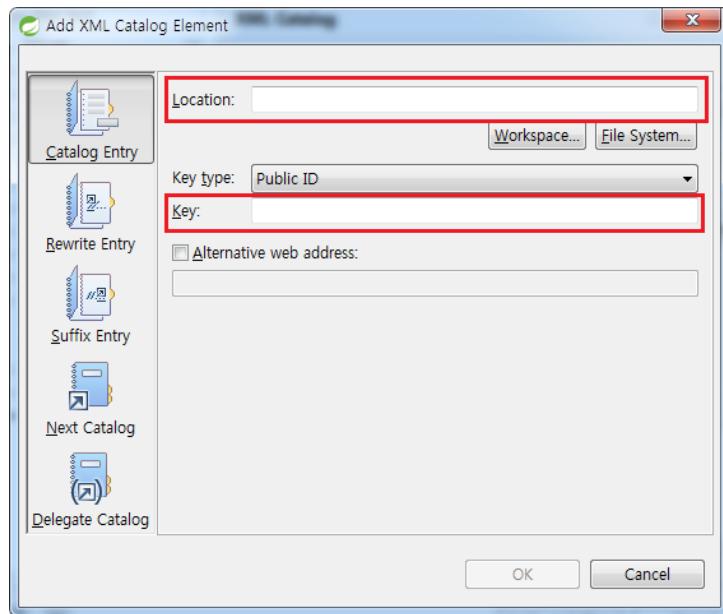
다음의 내용을 차례로 입력하여 추가한다.

Location : skiaf-template/src/main/resources/static/dtd/mybatis-3-config.dtd

Key : -//mybatis.org//DTD Config 3.0//EN

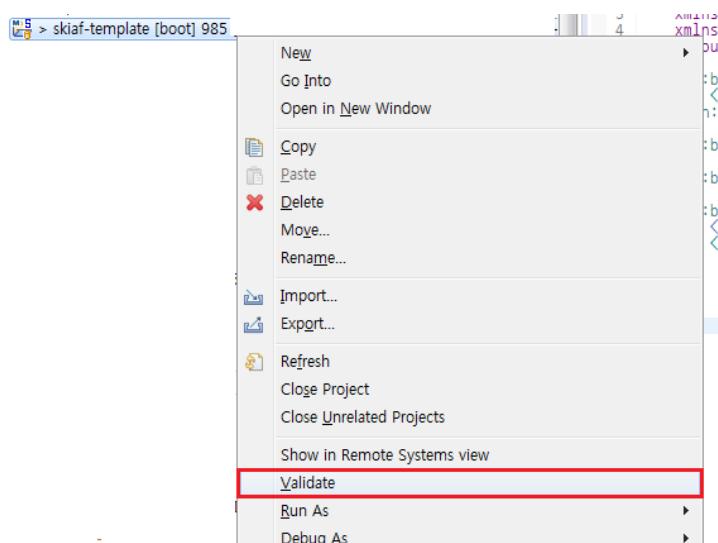
Location : skiaf-template/src/main/resources/static/dtd/mybatis-3-mapper.dtd

Key : -//mybatis.org//DTD Mapper 3.0//EN



#### [ DTD Validation 하기 ]

추가 후 프로젝트 우 클릭 - Validate 를 클릭한다.



#### I.4.4 Code Template

##### [ Code Template 파일 ]

같은 형식의 주석템플릿을 사용하기 위한 Code Template 파일이다.

하단의 내용을 메모장 또는 다른 문서저장 프로그램을 이용해 codetemplates.xml 파일로 저장한다.

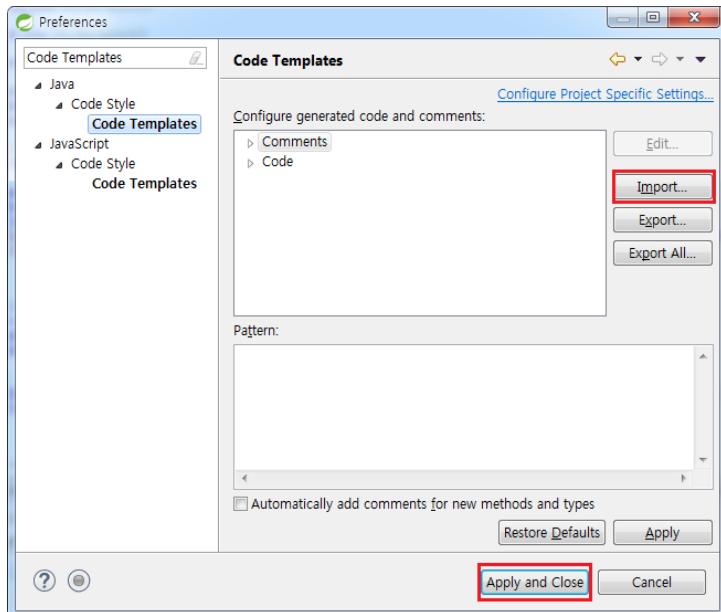
##### codetemplates\_skiaf\_v1.1.xml

```
<templates>
<template autoinsert="false" context="gettercomment_context" deleted="false" description="Comment for getter method"
enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.gettercomment" name="gettercomment"/>
<template autoinsert="false" context="filecomment_context" deleted="false" description="Comment for created Java files"
enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.filecomment" name="filecomment"/>*** */</template>
<template autoinsert="false" context="overridecomment_context" deleted="false" description="Comment for overriding
methods" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.overridecomment" name="overridecomment"/>*** <pre> *
</pre> * ${see_to_overridden} */</template>
<template autoinsert="false" context="fieldcomment_context" deleted="false" description="Comment for fields" enabled="true"
id="org.eclipse.jdt.ui.text.codetemplates.fieldcomment" name="fieldcomment"/>*** <pre> * </pre> */</template>
<template autoinsert="false" context="methodcomment_context" deleted="false" description="Comment for non-overriding
methods" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.methodcomment" name="methodcomment"/>*** <pre> *
</pre> */</template>
<template autoinsert="false" context="typecomment_context" deleted="false" description="Comment for created types"
enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.typecomment" name="typecomment">
*** <pre> * * History * - ${date} | ${user} | 최초작성. * </pre> */
</template>
<template autoinsert="false" context="constructorcomment_context" deleted="false" description="Comment for created
constructors" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.constructorcomment" name="constructorcomment"/>*** <pre> * </pre> */</template>
<template autoinsert="false" context="delegatecomment_context" deleted="false" description="Comment for delegate methods"
enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.delegatecomment" name="delegatecomment"/>*** <pre> * </pre> *
${see_to_target} */</template>
<template autoinsert="false" context="settercomment_context" deleted="false" description="Comment for setter method"
enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.settercomment" name="settercomment"/>
</templates>
```

##### [ 설정 ]

STS - Window - Preferences - Java - Code Style - Code Template - Import 클릭

codetemplate.xml 파일을 Import 한 후, Apply and Close 버튼을 눌러 창을 닫는다.



#### [ 사용하기 ]

Alt + Shift + J 를 누르면 적용된 주석이 자동으로 생성된다.

### I.4.5 Indentation (들여쓰기)

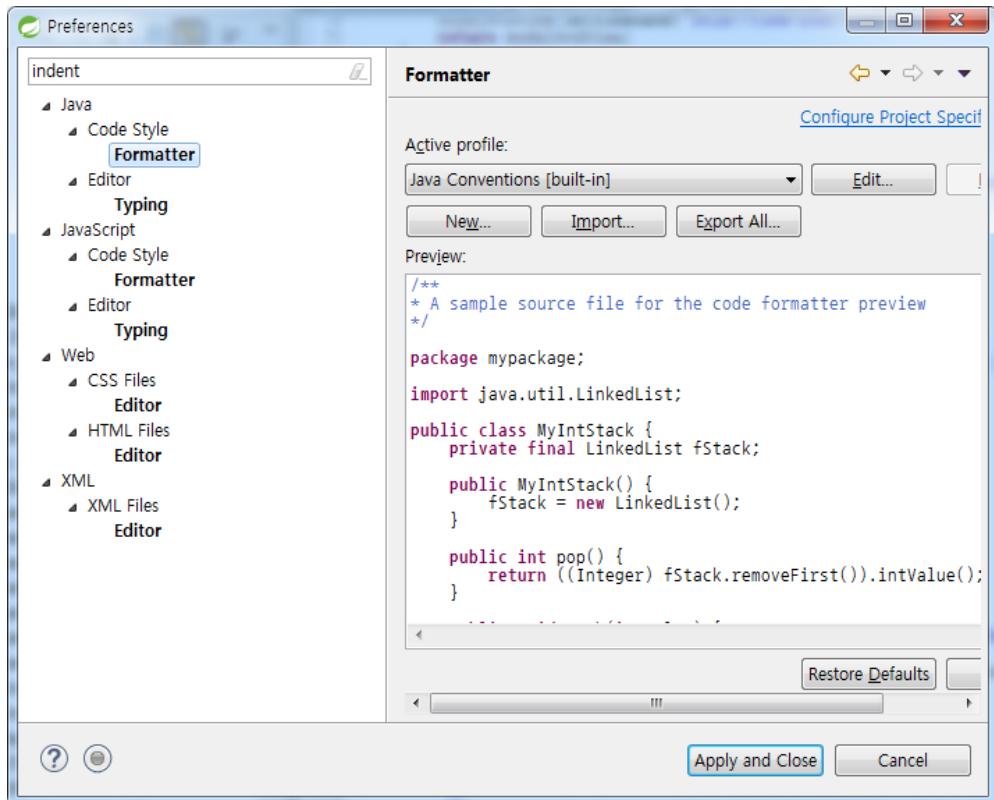
#### [ Indentation ]

기본 들여쓰기는 Tab 이지만 skiaf-template의 기본 들여쓰기는 4번 띄어쓰기(space 4번)이다.

Tab을 사용할 경우, 개발자 환경에 따라 Indentation이 깨지는 경우가 있으므로, space 문자를 사용하는 것을 권장한다.

#### [ Indentation 설정하기 - Java ]

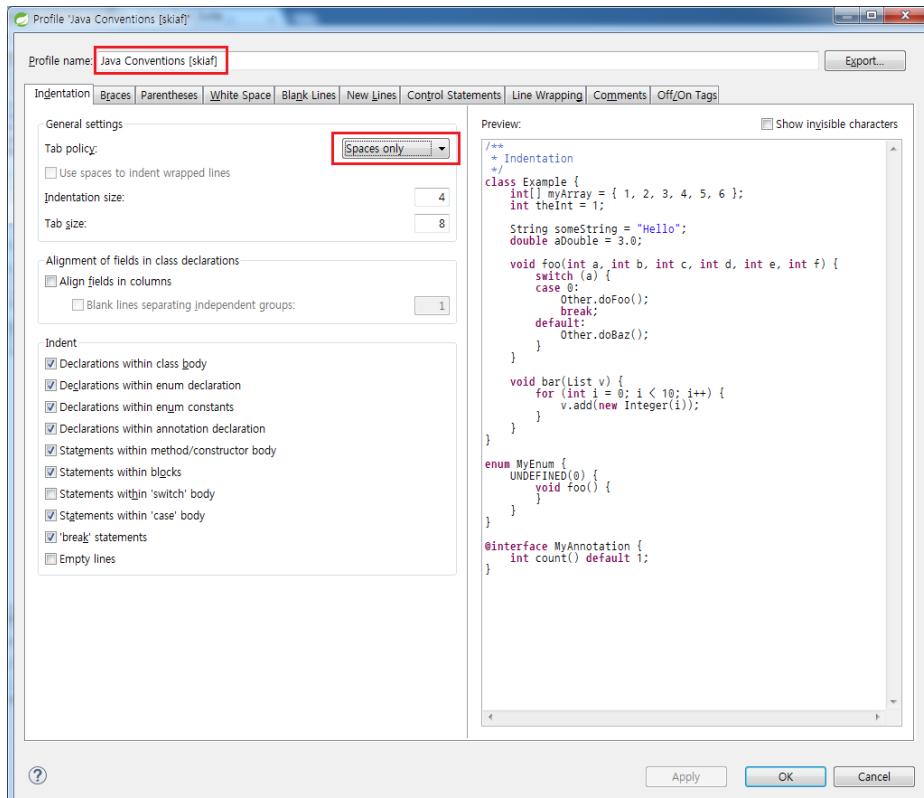
STS - Window - Preferences - Java - Code Style - Formatter



Java Conventions [built-in] - Edit 버튼 클릭

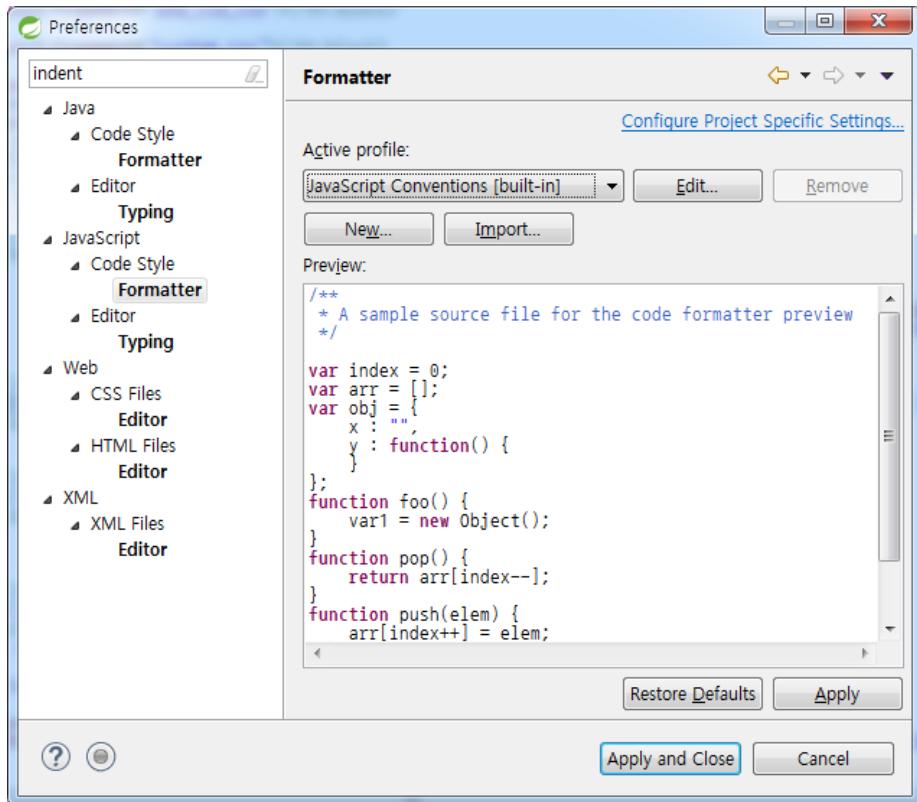
Profile name - Java Conventions [skiaf]로 변경

Tab policy - Spaces Only로 변경



### [ Indentation 설정하기 - JavaScript ]

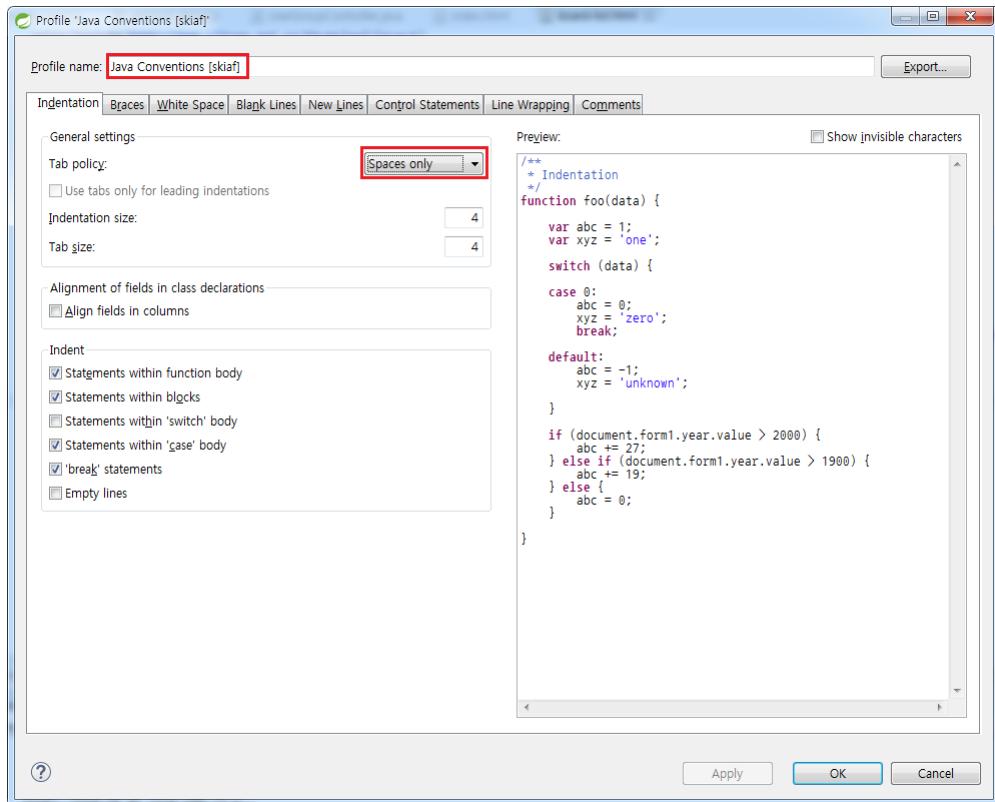
STS - Window - Preferences - JavaScript - Code Style - Formatter



Java Conventions [built-in] - Edit 버튼 클릭

Profile name - Java Conventions [skiaf]로 변경

Tab policy - Spaces Only로 변경



## [ Indentation 사용하기 ]

STS의 자동 들여쓰기 단축키는 Ctrl + Shift + F 이다.

## [ HTML, CSS의 Indentation 설정하기 ]

Atom에서 설정할 경우 : File - settings 화면에서 Editor 탭을 선택하고 Tab Length 를 변경한다.

SKIAF의 기본 들여쓰기에 맞춰서 4로 설정한다.

**Settings**

Core

**Editor**

URI Handling

System

Keybindings

Packages

Themes

Updates

Install

Open Config Folder

Inserted when the Tab key is pressed.

**Soft Wrap**  
Wraps lines that exceed the width of the window. When **soft wrap At Preferred Line Length** is set, it will wrap to the number of characters defined by the **Preferred Line Length** setting.

**Soft Wrap At Preferred Line Length**  
Instead of wrapping lines to the window's width, wrap lines to the number of characters defined by the **Preferred Line Length** setting. This will only take effect when the soft wrap config setting is enabled globally or for the current language. Note: If you want to hide the wrap guide (the vertical line) you can disable the **wrap-guide** package.

**Soft Wrap Hanging Indent**  
When soft wrap is enabled, defines length of additional indentation applied to wrapped lines, in number of characters.  
Default: 0

**Tab Length**  
Number of spaces used to represent a tab.  
4

**Tab Type**  
Determine character inserted when Tab key is pressed. Possible values: "auto", "soft" and "hard". When set to "soft" or "hard", soft tabs (spaces) or hard tabs (tab characters) are used. When set to "auto", the editor auto-detects the tab type based on the contents of the buffer (it uses the first leading whitespace on a non-comment line), or uses the value of the Soft Tabs config setting if auto-detection fails.  
auto

**Undo Grouping Interval**  
Time interval in milliseconds within which text editing operations will be grouped together in the undo history.  
Default: 300

**Zoom Font When Ctrl Scrolling**  
Change the editor font size when pressing the Ctrl key and scrolling the mouse up/down.

### [ HTML, CSS의 Indentation 사용하기 ]

Atom에서의 Indent 단축키는 **Ctrl + ]** 이고 메뉴에서 **Edit - Lines - Auto Indent**를 사용해도 된다.

### I.4.6 플러그인 설정

#### 참고사항

skiaf-template에서 STS 플러그인을 따로 제공하지 않기 때문에 아래의 플러그인 목록을 확인하고 각자 설치 해줘야한다.

이 플러그인은 skiaf-template를 개발하면서 사용한 플러그인들이고 그 외 다른 플러그인을 사용해도 무방하다.

## [ 플러그인 목록 ]

- Properties Editor (선택)
  - 자바 properties 파일의 기본 인코딩은 ISO-8859-1(ASCII 코드 기반의 확장 인코딩)이다. EUC-KR이나 UTF-8 인코딩 상태의 한글로 입력을 하게되면 실제 웹 페이지에서 프로퍼티 파일의 문자열을 참조할 경우 모두 깨진 문자열 형태로 출력되게 된다.  
따라서, 한글이나 다른 유니코드 문자를 정상적으로 출력하기 위해 ASCII값으로 일치 시켜주는 작업을 할 수 있게 해주는 플러그인 이다.
  - <http://marketplace.eclipse.org/content/properties-editor>
- SQL Development Tools (선택)
  - 데이터베이스 관리, SQL 작성 및 실행을 위한 플러그인이다. 다양한 데이터베이스의 디버깅을 가능하게 하는 확장 기능을 제공한다.
  - <http://marketplace.eclipse.org/content/sql-development-tools>
- Subversive - SVN Team Provider (선택)
  - SVN 협상관리를 하기 위한 플러그인이다. SVN의 업데이트, 커밋, 변경 기록을 보고 다른 작업을 직접 수행할 수 있다.
  - <http://marketplace.eclipse.org/content/subversive-svn-team-provider>
- ERMaster (선택)
  - STS내에서 ERD를 만들 수 있는 플러그인이다. 기본기능은 모두 갖추고 있고, XML로 저장되기 때문에 버전관리에도 용이하다.
  - <https://sourceforge.net/projects/ermaster/files/>

## I.5. Nexus 설정

외부 인터넷 접속이 가능한 환경에서 개발하는 경우에는 Nexus 연동이 필요없으며, VDI 내부 (외부 인터넷 차단)에서 개발하는 경우에는 SKI 전용 Nexus를 이용한다.

SKI 전용 Nexus에 대한 관리는 에너지/화학 지능화O.G에서 담당하고 있으므로 추가문의는 해당 운영자에게 문의한다.

- Nexus 서버와 개발자 PC간 방화벽 연결여부는 사전에 확인이 필요하다.
- Devops 적용 매뉴얼을 참고해야 한다.

## [ Nexus 설정 ]

프로젝트의 pom 파일을 수정할 필요없이, 아래의 파일만 경로에 추가하면 된다.

C:\Users\{사용자 ID}\.m2\settings.xml

### Setting.xml

```
<settings>
  <servers>
    <server>
      <id>central</id>
    </server>
  </servers>
  <mirrors>
    <mirror>
      <!--This sends everything else to /public -->
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <!-- refers to SKI-oriented nexus repository -->
      <url>http://10.178.13.150:8987/repository/maven-public/</url>
    </mirror>
  </mirrors>
</settings>
```

maven-public은 ski-maven-central과 maven-releases (3rd lib용)을 그룹으로 묶은 repository를 말한다.

기본적으로 ski-maven-central을 사용하나, maven-central에서 제공해주지 않는 외부 라이브러리 (예: ojdbc8, mssql 등)는 maven-release에 업로드되어 있다.

### [ Maven 라이브러리 추가 ]

maven-central에 없는 3rd-party 라이브러리를 추가 사용할 경우, Nexus 서버의 "maven-release" repository에 업로드되도록 운영자 (에너지/화학지능화 O.G)에게 요청한다.

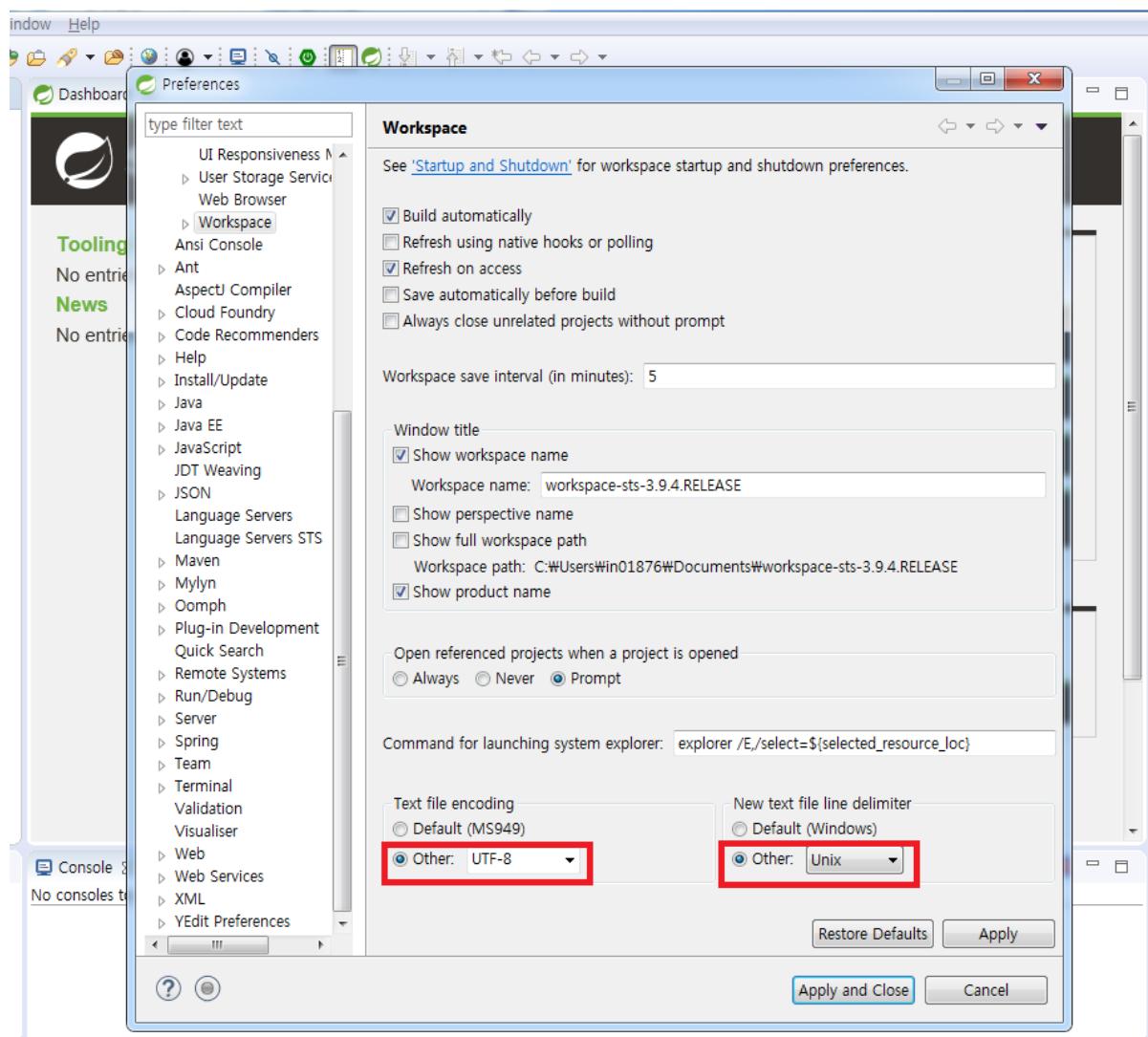
그룹웨어 포털 서버의 SSO인증기능과 연동하기 위한 jsso-agent-jdk14.jar와 같은 custom lib도 "maven-release"에 업로드되어 있다.

	Name ↑	Type	Format	Status
	maven-public	group	maven2	Online
	maven-releases	hosted	maven2	Online
	maven-snapshots	hosted	maven2	Online
	nuget-group	group	nuget	Online
	nuget-hosted	hosted	nuget	Online
	nuget.org-proxy	proxy	nuget	Online - Ready to Connect
	ski-maven-central	proxy	maven2	Online - Remote Available

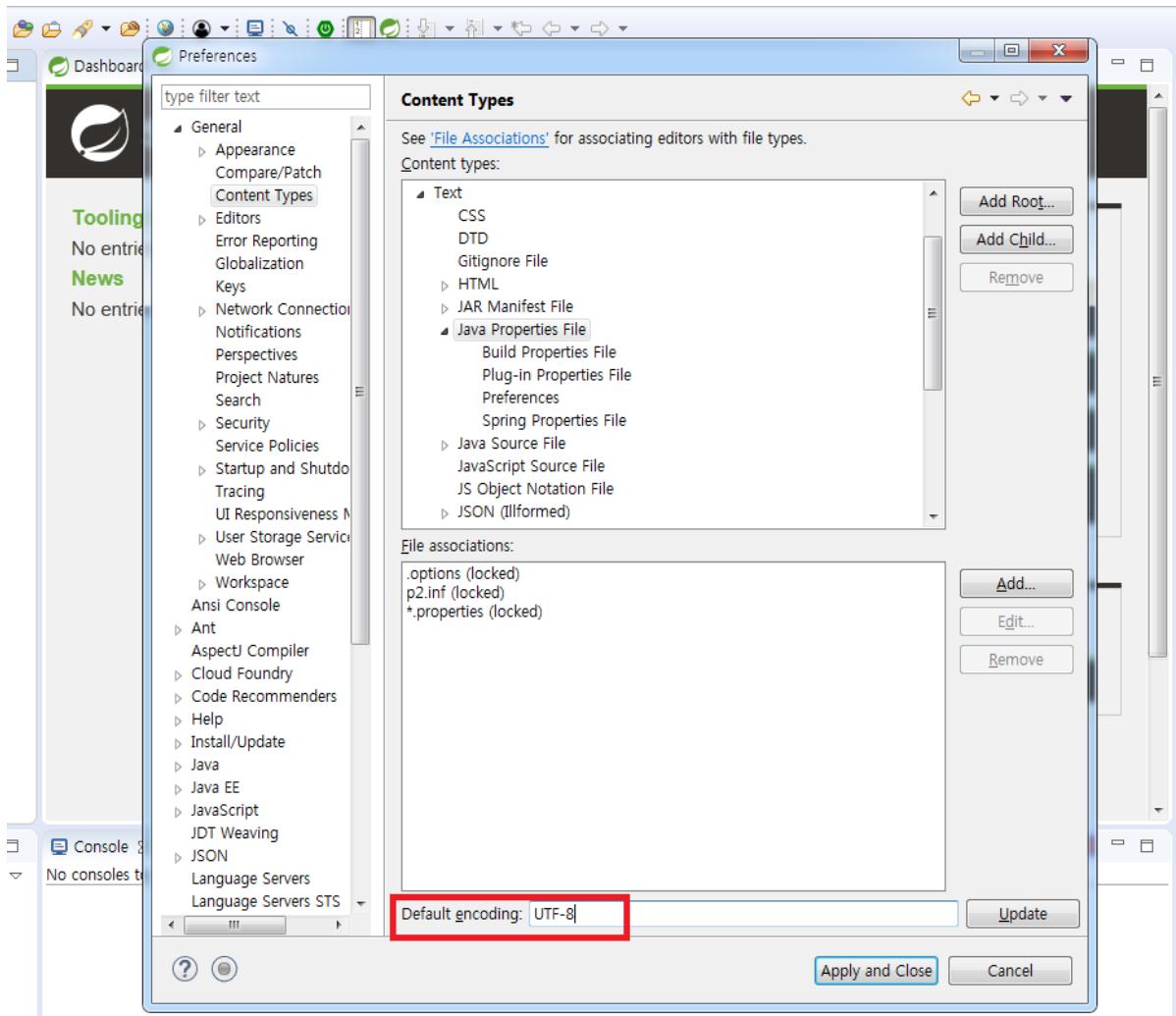
## I.6. 개발환경 설정

### [ Encoding - Java ]

- STS에서 상단메뉴중 Window → Preferences 클릭후, 팝업창에서 왼쪽 메뉴 General → Workspace선택 후 값을 Text file encoding : UTF-8 , New text file line delimiter : Unix 로 세팅 후 [Apply] 클릭하여 저장



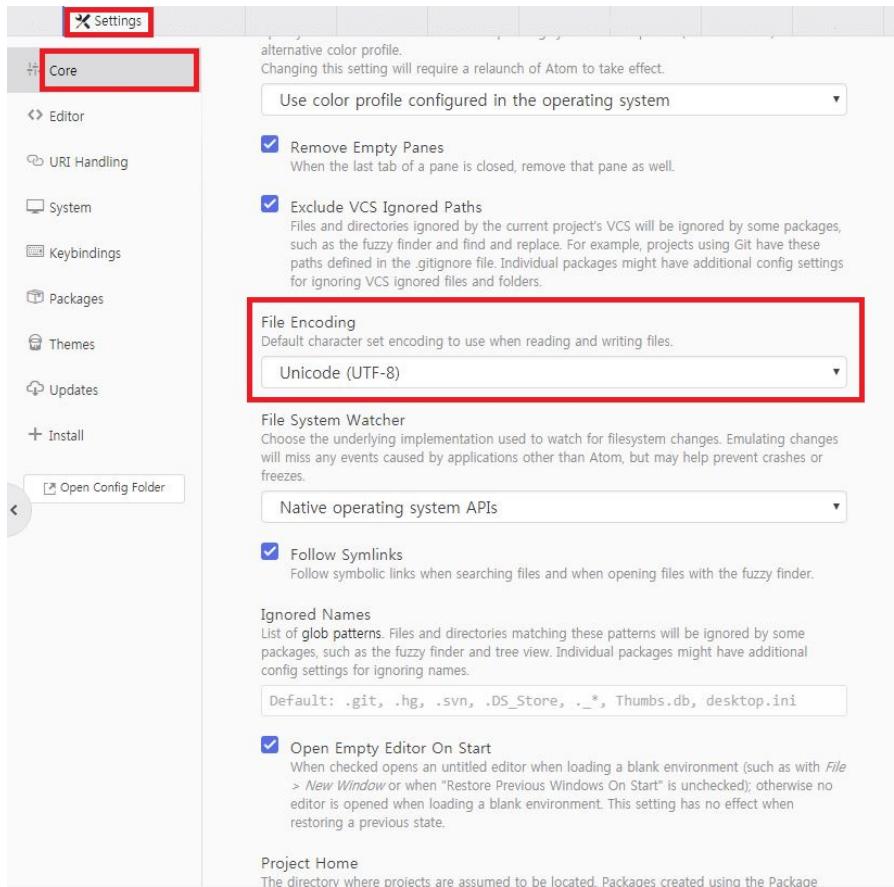
- STS에서 상단메뉴중 Window → Preferences 클릭후, 팝업창에서 왼쪽 메뉴 General → Content Types → Java Properties File선택 후 값을 Default encoding : UTF-8 로 세팅 후 [Apply] 클릭하여 저장



- JavaScript, XML 파일은 기본적으로 UTF-8로 설정 되어 있다.

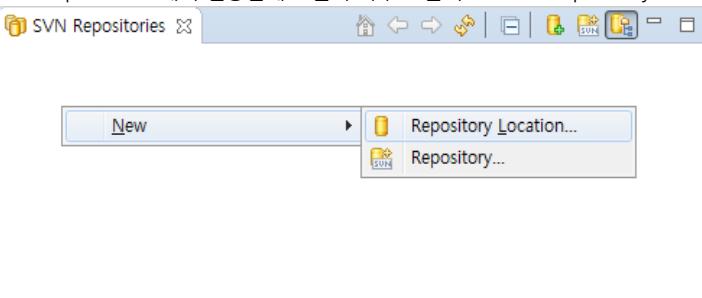
### [ Encoding - HTML, CSS ]

- Atom에서 HTML, CSS의 Default encoding 설정 : File - settings 화면에서 Core 탭을 선택하고 File Encoding을 UTF-8로 변경한다.

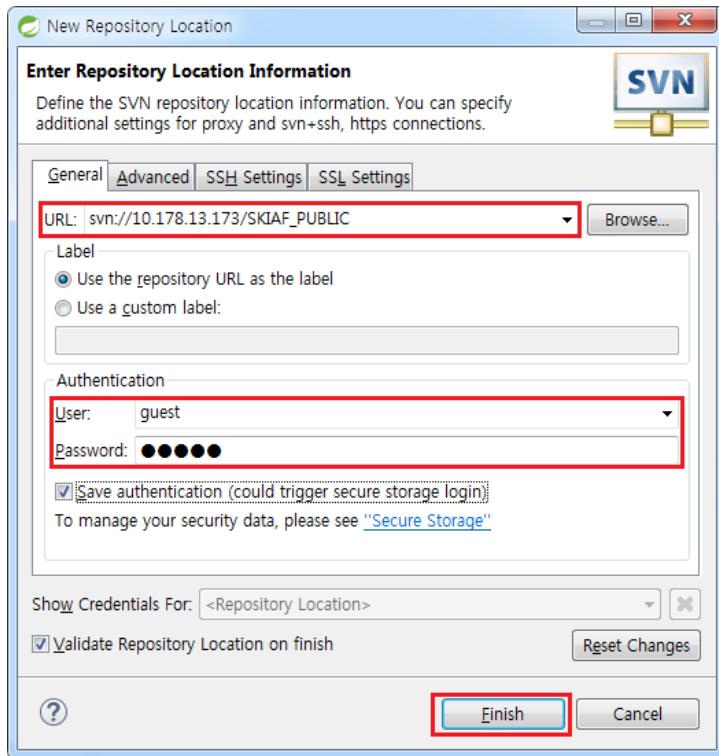


## [ SVN에서 Export 하기 ]

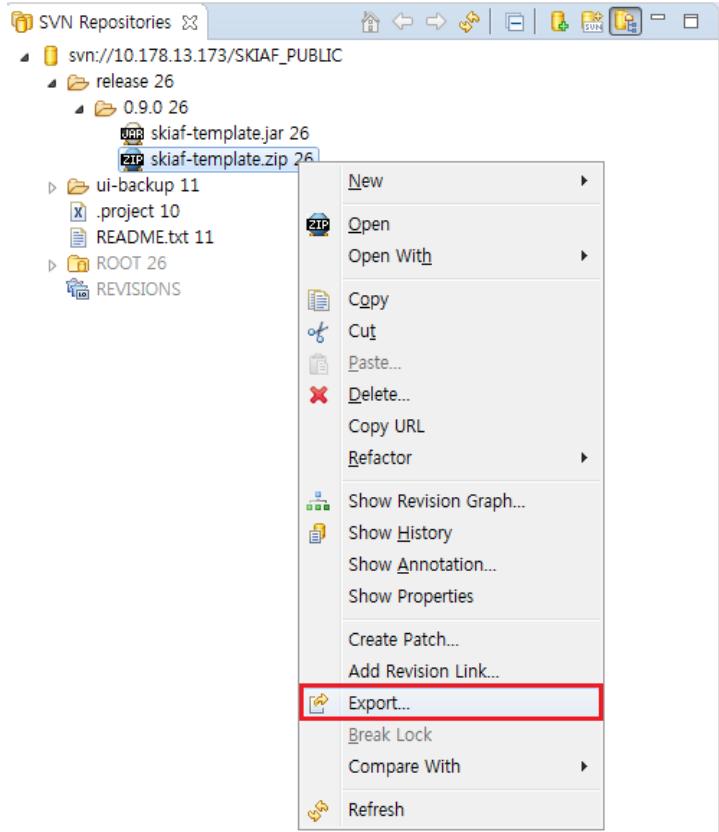
- SVN Repositories에서 빈공간에 오른쪽 마우스 클릭 - New - Repository Location 클릭



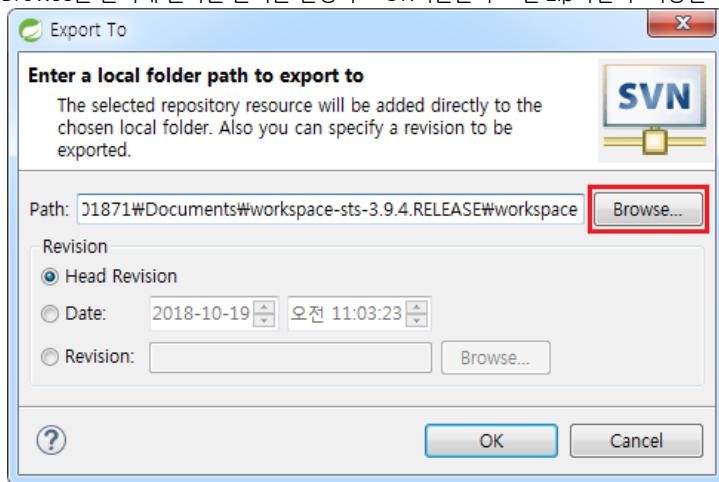
- SVN URL과 계정명 비밀번호를 입력하고 Finish를 클릭
- SVN URL : svn://svn.skinnovation.com/SKIAF\_PUBLIC ( guest / guest )
- 대소문자 구분해서 정확하게 입력해야 한다.



- SVN URL - release - 버전 - skiaf-template.zip 오른쪽 마우스 클릭 - Export 클릭

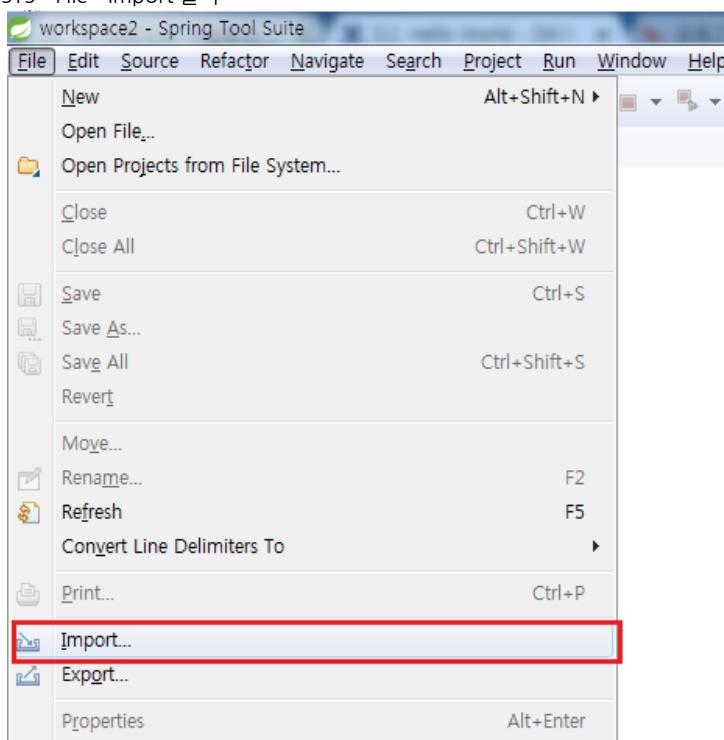


- Browse를 클릭해 원하는 폴더를 설정하고 OK버튼을 누르면 zip파일이 저장된다.

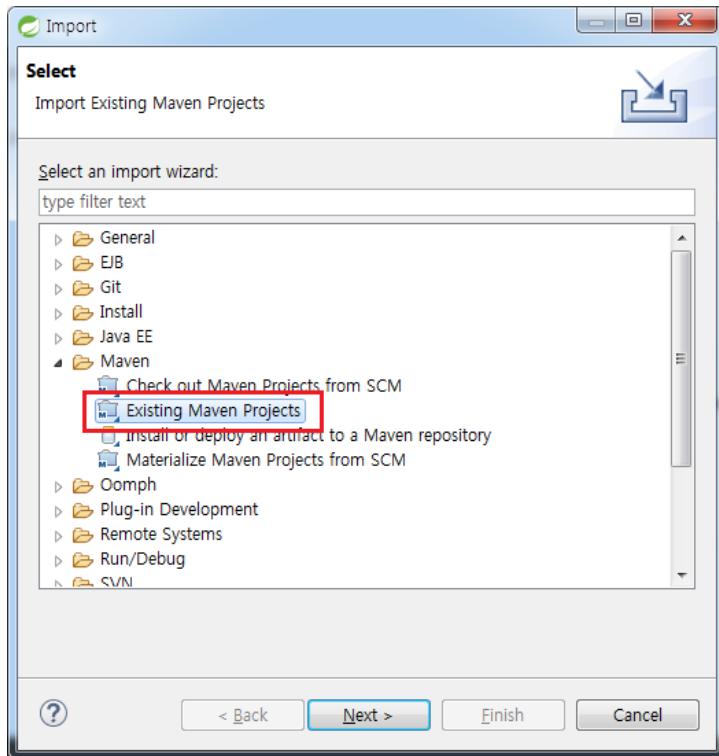


### [ STS에 Import 하기 ]

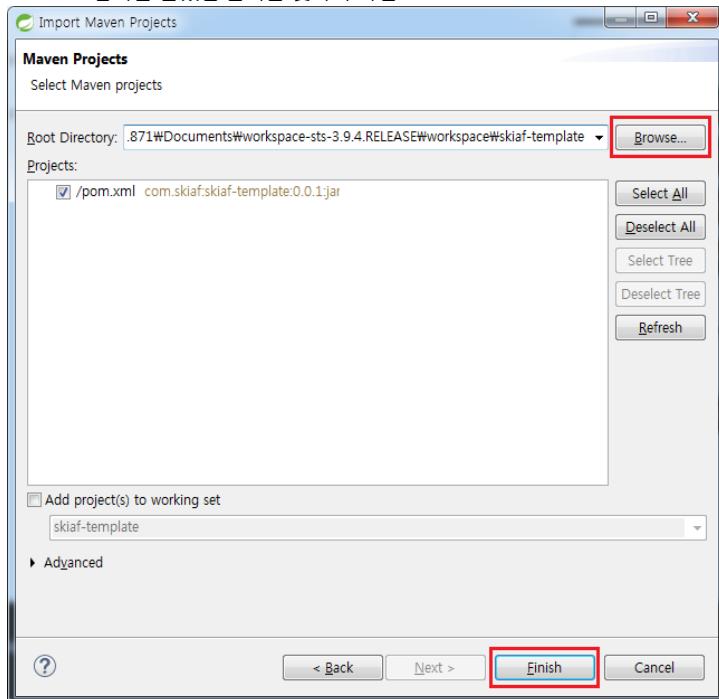
- Export 받은 skiaf-template.zip 파일을 사용할 workspace에 압축풀기
- STS - File - Import 클릭



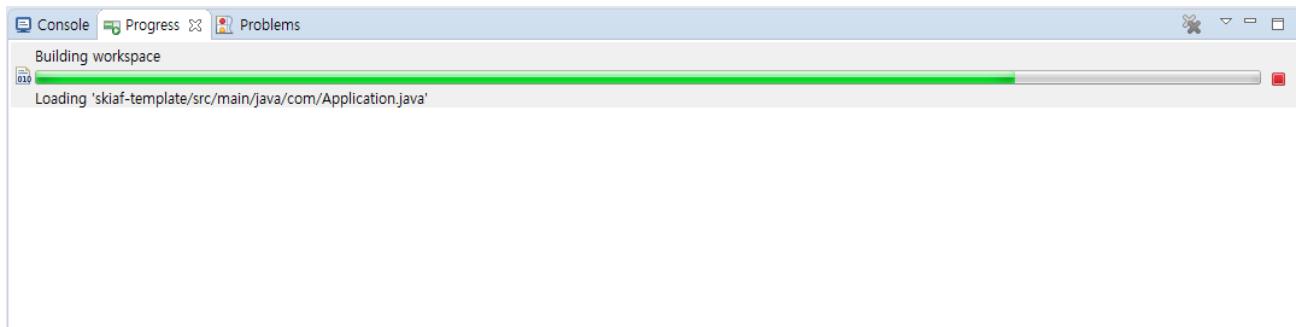
- Maven - Existing Maven Projects - Next



- Browse - 압축을 풀었던 폴더를 찾아서 확인 - Finish

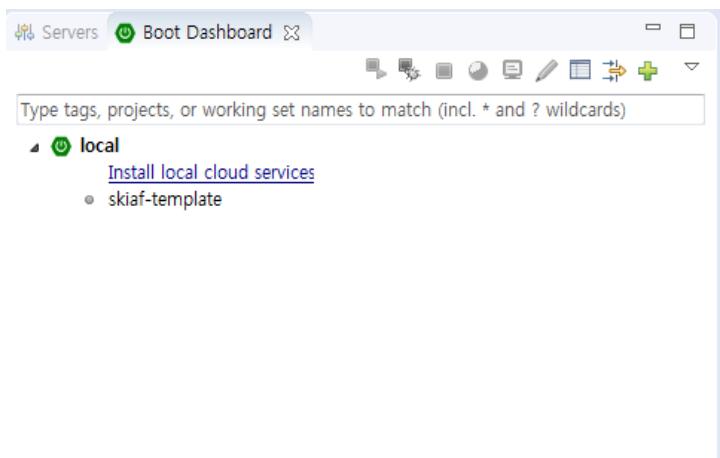
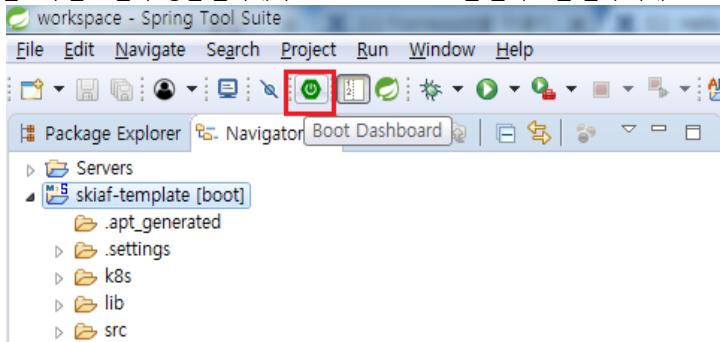


- Import한 프로젝트가 자동으로 빌드될때 까지 기다린다. 빌드가 되는 과정은 Progress 탭에서 확인 가능하다.

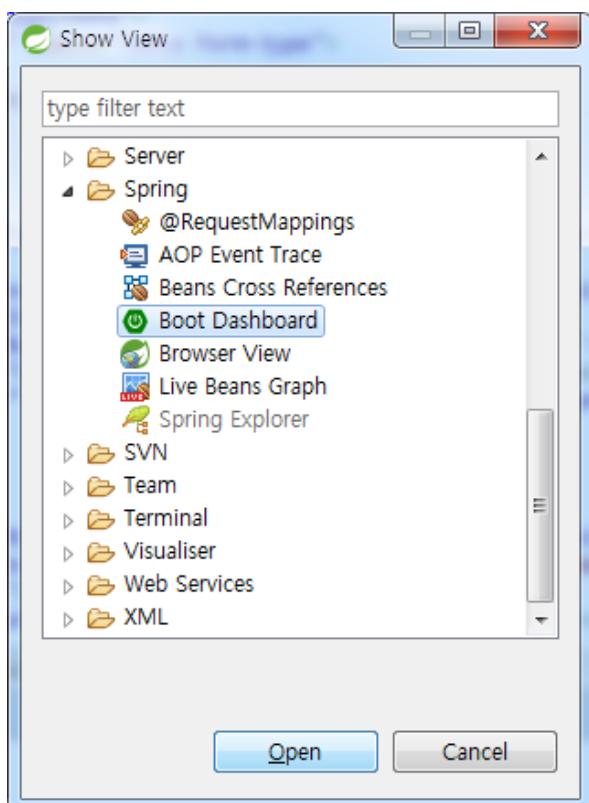
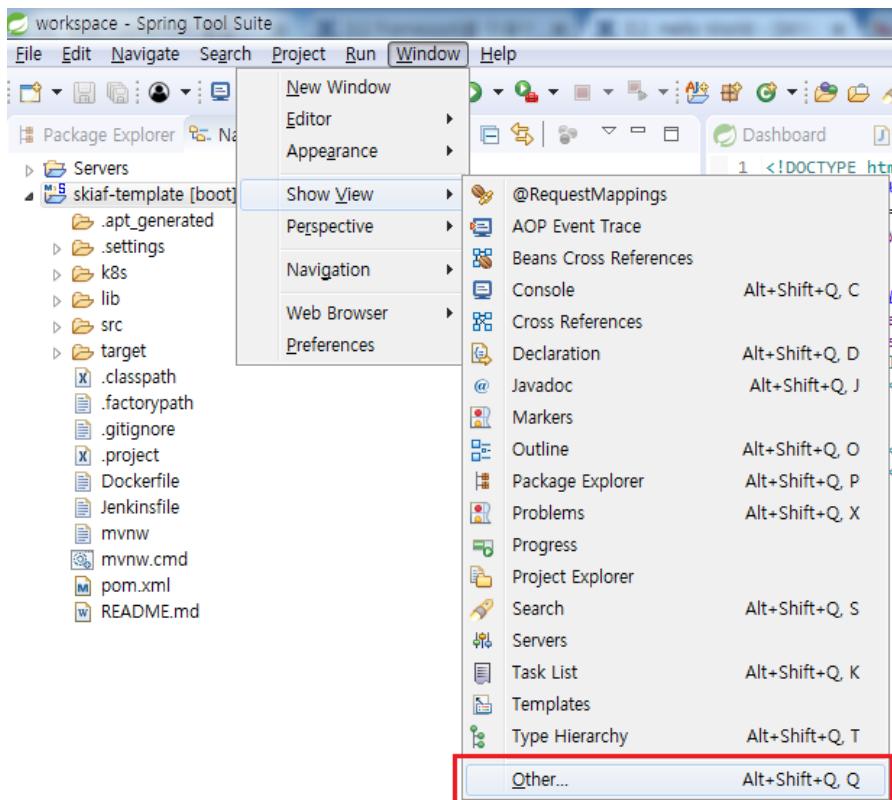


### [ template 구동하기 ]

- 빌드가 완료 된 후 상단 툴바에서 Boot Dashboard 를 클릭 또는 왼쪽 아래 Boot Dashboard 탭을 확인한다.

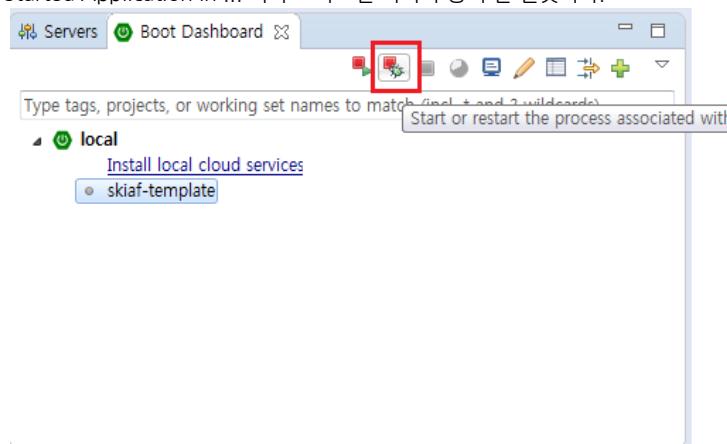


- Boot Dashboard가 없을 경우  
Window - Show View - Other 클릭  
Spring - Boot Dashboard - Open 하면 창을 볼 수 있다.



- 서버를 구동하려면 Boot Dashboard에서 skiaf-template을 선택하고 Start debug mode를 클릭한다. Console 창에서 구동여부를 확인할 수 있다.

Started Application in ... 이라고 나오면 서버 구동이 잘 된것이다.



```

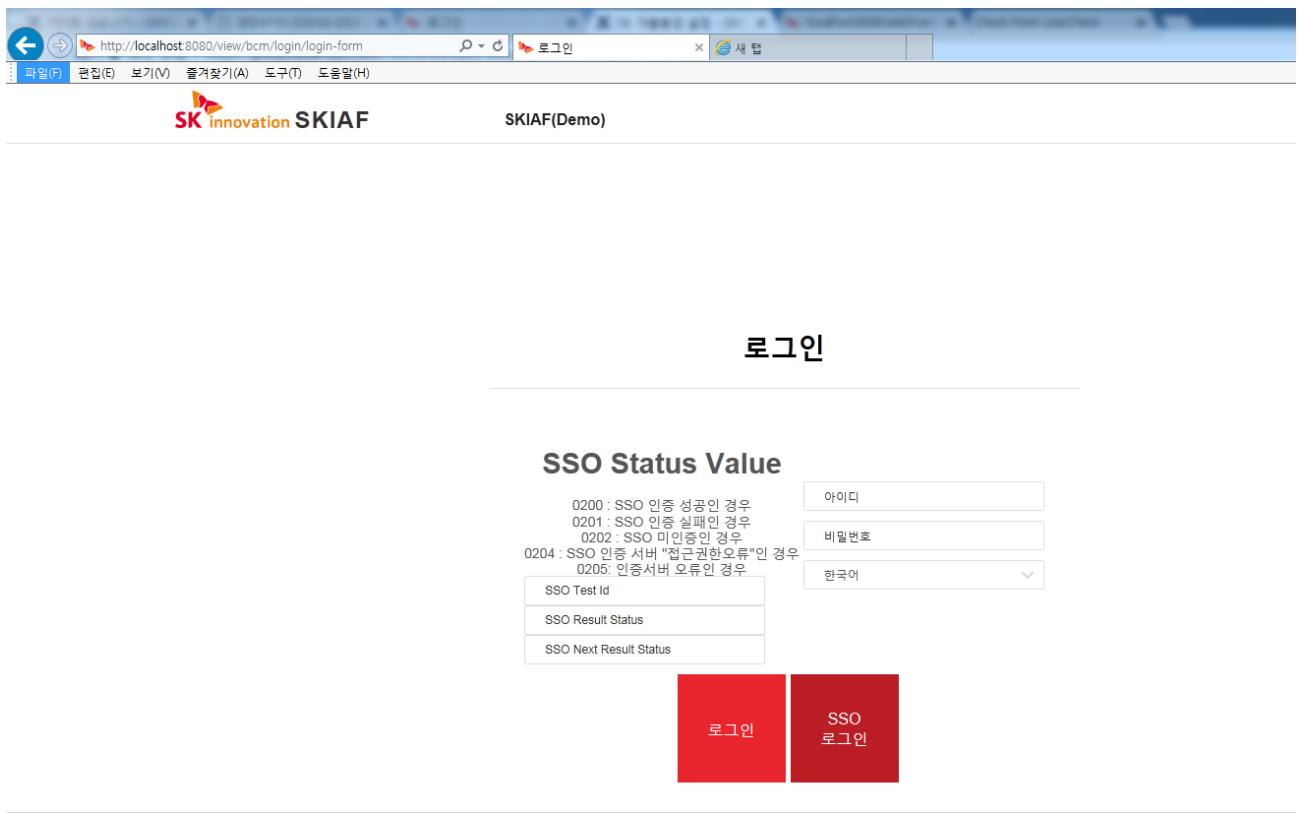
Console Progress Problems
skiaf-template - Application [Spring Boot App] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 10. 10. 오후 5:15:46)
2018-10-10 17:16:26.219 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: duplicateCheckU
2018-10-10 17:16:26.222 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: findOneUsingGET
2018-10-10 17:16:26.225 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: findOneDetailUs
2018-10-10 17:16:26.233 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: findQueryByKeyw
2018-10-10 17:16:26.247 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: updateUsingPUT_
2018-10-10 17:16:26.254 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: createUsingPOST
2018-10-10 17:16:26.258 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: duplicateCheckU
2018-10-10 17:16:26.260 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: findOneUsingGET
2018-10-10 17:16:26.265 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: findQueryByKeyw
2018-10-10 17:16:26.270 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: updateUsingPUT_
2018-10-10 17:16:26.273 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsersByUs
2018-10-10 17:16:26.277 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsersByUser
2018-10-10 17:16:26.283 INFO 8972 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: activeRoleListU
2018-10-10 17:16:26.397 INFO 8972 --- [           main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
2018-10-10 17:16:26.404 INFO 8972 --- [           main] com.skiaf.demo.cache.CacheManagerCheck : 

=====
Using cache manager: org.springframework.cache.ehcache.EhCacheCacheManager
=====

2018-10-10 17:16:26.442 INFO 8972 --- [           main] com.Application : Started Application in 36.725 seconds (JVM running

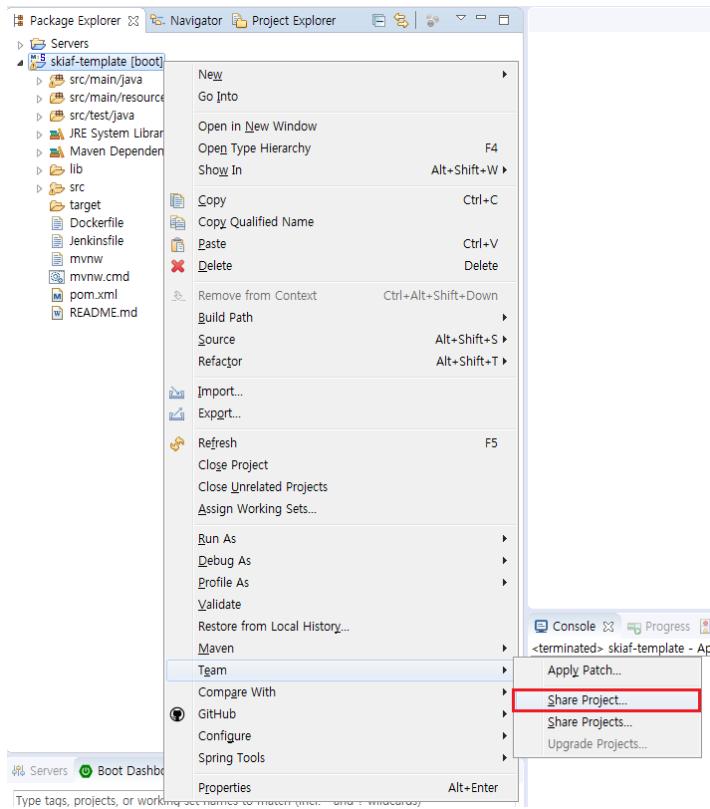
```

- 브라우저에서 확인하려면  
브라우저 주소창에  
`http://localhost:8080/`을 입력하면 된다.
- 서버가 정상적으로 작동된다면 다음과 같은 화면을 확인할 수 있다.

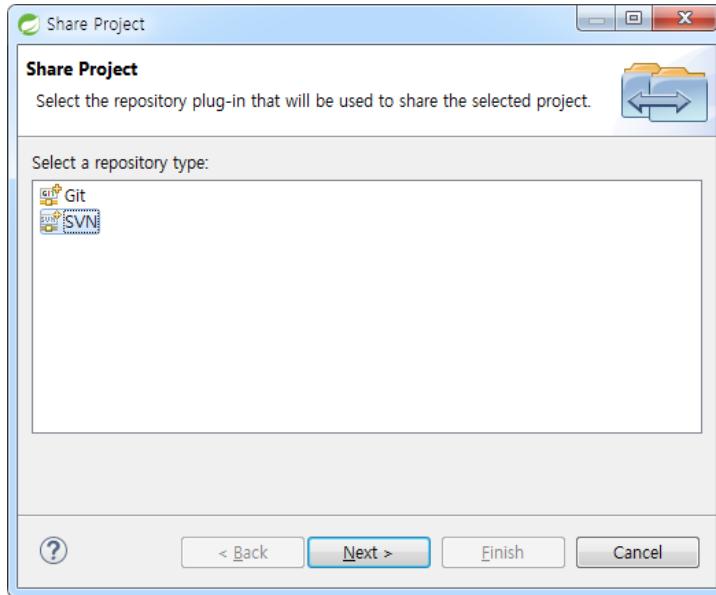


## [ SVN 연결하기 ]

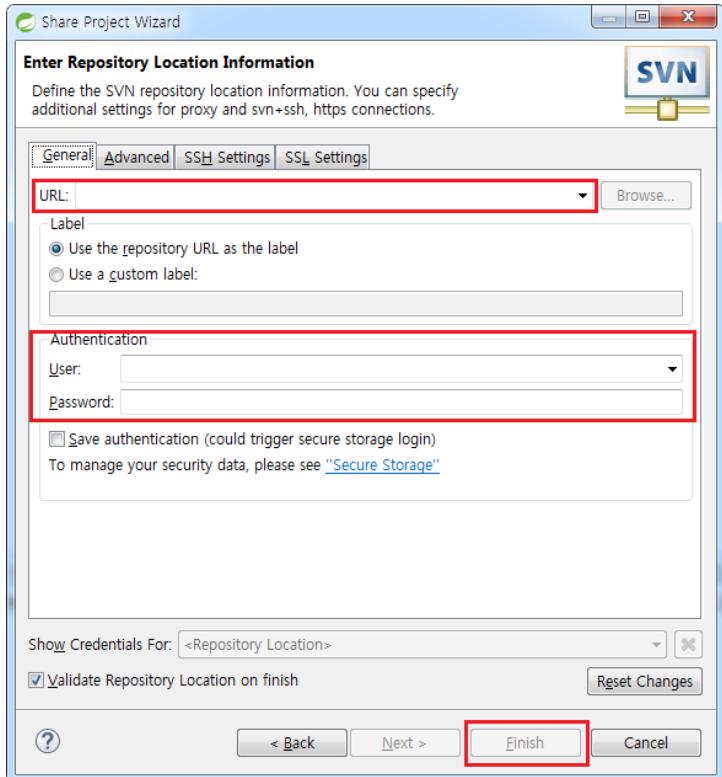
- Import한 프로젝트 오른쪽 마우스 클릭 - Team - Share Project 클릭



- SVN을 선택

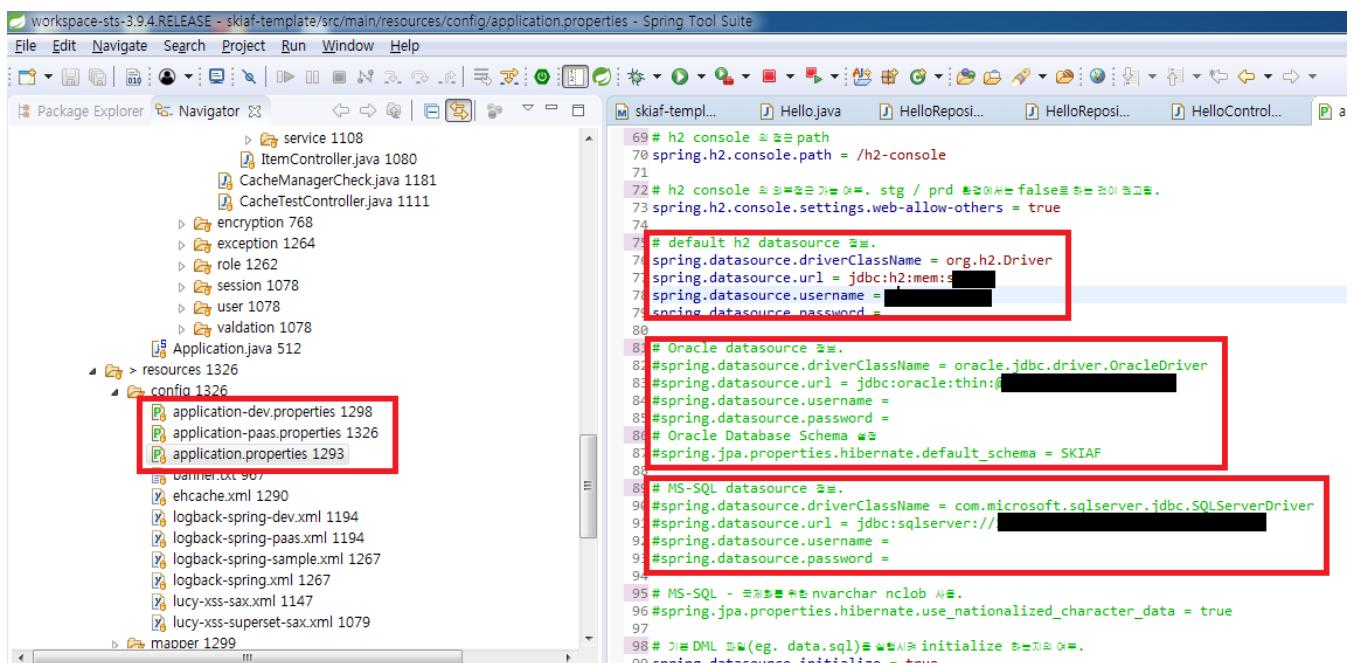


- 형상관리 서비스 관리자에게 받은 SVN URL과 user, password를 입력하고 Finish를 누르면 연결이 완료된다.



## [ DB Setting ]

SKIAF Framework는 application.properties 파일내에서 DB를 설정을 바꿔서 H2, Oracle, MS-SQL중 하나를 선택하여 연결할 수 있다  
현재세팅은 로컬에서 H2 Database를 사용하게 되어 있다.



\* H2 DB 설정 예시

### H2 DB

```
# h2 console 사용여부. stg / prd 환경에서는 false로 하는 것이 권고됨.  
spring.h2.console.enabled = true  
  
# h2 console 의 접근 path  
spring.h2.console.path = /h2-console  
  
# h2 console 의 외부접근 가능 여부. stg / prd 환경에서는 false로 하는 것이 권고됨.  
spring.h2.console.settings.web-allow-others = true  
  
# default h2 datasource 정보.  
spring.datasource.driverClassName = org.h2.Driver  
spring.datasource.url = jdbc:h2:mem:skiaf  
spring.datasource.username =  
spring.datasource.password =
```

\* Oracle DB 설정 예시

### Oracle DB

```
# Oracle datasource 정보.  
#spring.datasource.driverClassName = oracle.jdbc.driver.OracleDriver  
#spring.datasource.url = jdbc:oracle:thin:@오리클DB IP주소:포트번호:SID 이름  
#spring.datasource.username =  
#spring.datasource.password =
```

\* Microsoft SQL 설정 예시

### MS-SQL

```
# MS-SQL datasource 정보.  
#spring.datasource.driverClassName = com.microsoft.sqlserver.jdbc.SQLServerDriver  
#spring.datasource.url = jdbc:sqlserver://MS-SQL IP주소;databaseName=DB이름  
#spring.datasource.username =  
#spring.datasource.password =
```