

그래봐야 코딩, 그래도 코딩

Home ETC ▾ Infra ▾ Joy ▾ Programing ▾ Tool ▾ Web ▾ Search

INFRA > JENKINS

FOLLOW:



빠른 코딩 입문과 코딩실력 향상

코딩 입문은 더 쉽게, 빠르게, 더 간편하게 코딩어레이 키트가



centOs7의 Jenkins에서 Spring Maven 프로젝트(SVN연동) 빌드 후 Tomcat7에 배포하기

2018-05-23

#AUTO BUILD DEPLOY, #BUILD, #CENTOS7, #DEPLOY, #JENKINS, #SPRING MAVEN PROJECT, #TOMCAT7, #UNMAPPABLE CHARACTER FOR ENCODING UTF8, #WAR

📖 들어가기

이 실습은 로컬 개발pc(Wincows10 pro) + SVN 서버(CentOs7_1) + Tomcat 서버(CentOs_2) + Jenkins 서버(CentOs7_3) 환경에서 실습을 진행했다.(윈도우 pc 1개, centos7 3개)
실습을 위해 기본적으로 maven 프로젝트와 이 프로젝트를 관리하는 svn서버, tomcat 서버, 젠킨스 서버가 필요하다.

📖 실습환경

- CentOs7
- jdk1.8
- maven3

NEWER

폭포수 방법론의 단점



OLDER



CentOs7에 maven(메이븐) 설치하기

RECENTS

WEB > REACT

리엑트(REACT) WINDOWS10 개발환경 셋팅
2019-04-27

WEB > NODE

NODE 프로젝트 DEPENDENCIES 일괄 버전 업데이트
2019-04-11

PROGRAMING > TYPESCRIPT

TYPESCRIPT-WEBPACK으로 빌드하기
2019-03-21

JOY > GUITAR

GUITAR 카피 - DAFT PUNK
INSTANT CRUSH
2019-02-06

- Tomcat7
- Spring mvc maven project.war

[INFRA](#) > [DOCKER](#)

DOCKER- DOCKER TOOLBOX

기본 IP

2019-01-31

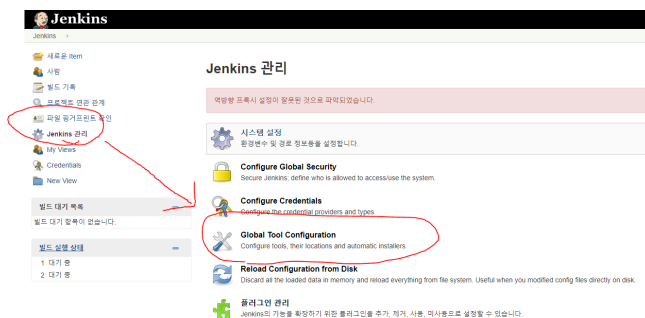
▶ 선행작업

[CentOs7에 jdk 설치](#)[CentOs7에 메이븐\(maven\) 설치](#)[CentOs7에 Jenkins설치](#)

▶ 젠킨스에 jdk, maven 연결

일단 젠킨스에서 maven 프로젝트를 빌드하려면 jdk, maven 을 연동해야 한다.

젠킨스 웹에서 Jenkins관리 - Global Tool Configuration을 클릭한다.



Jenkins관리 - Global Tool Configuration을 클릭 한다.

아래처럼 Global Tool Configuration화면이 나오는데 JDK, Maven항목에 버튼을 이용하여 각각 설치된 JDK, Maven 을 설정해야 한다.



CATEGORIES

▶ ETC (4)

▶ Hexo (2)

▶ Windows10 (2)

▶ Infra (30)

▶ Apache (1)

▶ CentOS (11)

▶ Docker (7)

▶ Git (3)

▶ error (1)

▶ Jenkins (1)

▶ Maven (2)

▶ Mysql (2)

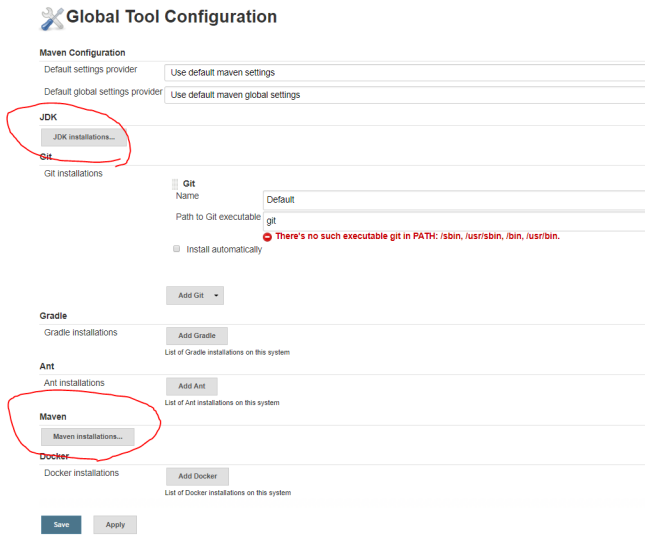
▶ Ubuntu (2)

▶ mongoDB (1)

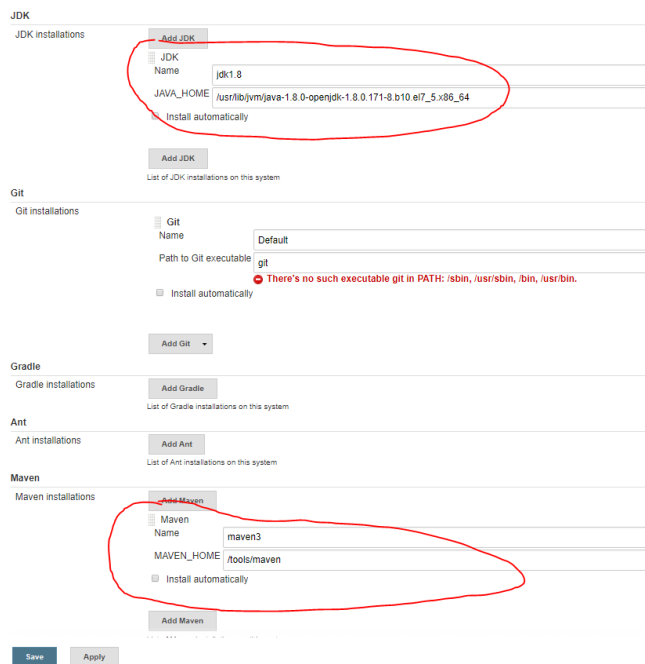
▶ Joy (28)

▶ Book (2)

▶ Guitar (1)



Global Tool Configuration 화면



Global Tool Configuration jdk, maven 설정

위 처럼 젠킨스 서버에 설치된 JDK 경로와,
Maven 경로를 설정하자.

Install automatically 항목을 끄자.(만약 설치가
않되어 있을 경우 자동 설치해주는 옵션이다.)
그리고 보면 알겠지만 jdk든 maven이던 여러버
전을 복수개 설정하여 젠킨스에서 유동적으로 골
라 사용할 수도 있다.

▶ 톰캣 배포용 플러그인 설치

▶ MotorBike (1)

▶ Music (21)

▶ Stuff (3)

▶ Programing (77)

▶ Algorithm (1)

▶ Android (7)

▶ Java (27)

▶ Code (2)

▶ Python (1)

▶ Spring (28)

▶ Error (1)

▶ Thymeleaf (5)

▶ tiles (1)

▶ SpringBoot (5)

▶ Typescript (1)

▶ 방법론 (3)

▶ 용어정리 (4)

▶ Tool (14)

▶ AndroidStudio (4)

▶ Atom (3)

▶ Chrome (1)

▶ IntelliJ (1)

▶ STS(Spring Tool Suite) (4)

▶ VisualStudioCode (1)

▶ Web (38)

▶ HTML (2)

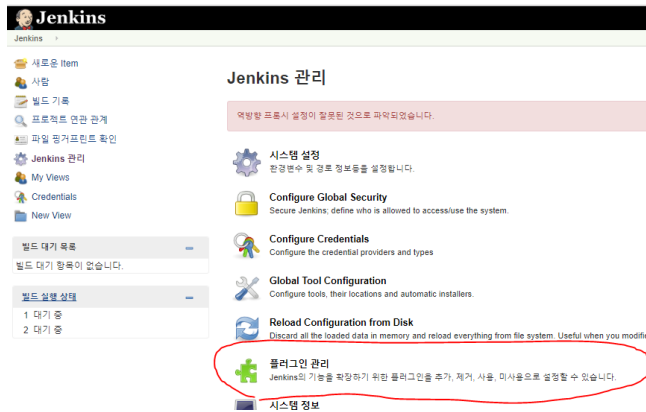
▶ Javascript (17)

▶ JQuery (1)

젠킨스가 빌드한 결과물(war파일)을 톰캣에게 배포하려면 Deploy to container Plugin 플러그인이 필요하다.

Deploy to container Plugin 플러그인을 설치하자.

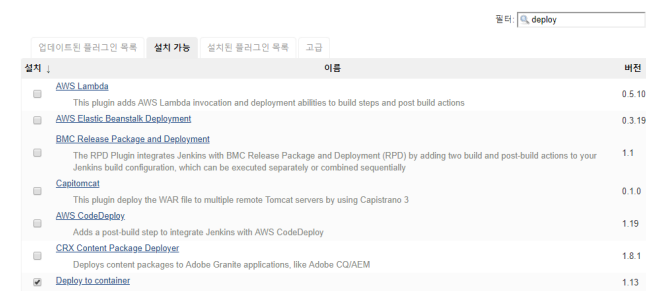
다시 jenkins 관리로 이동한 뒤, 플러그인 관리를 클릭하자.



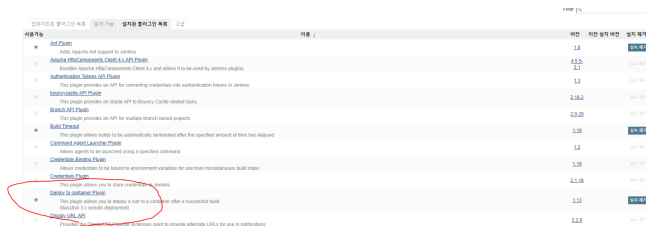
플러그인 관리로 이동

설치 가능 탭으로 이동하여 필터에 deploy를 검색하여 Deploy to container Plugin을 찾아 설치하자.

안전성을 위해 플러그인 설치시 자동 젠킨스 재시작 옵션을 추가하도록 하자.



플러그인 조회



플러그인 설치결과

- ▶ Node (5)
 - ▶ Express (3)
- ▶ React (1)
- ▶ Tip (2)
- ▶ Vue.js (5)
- ▶ Webpack (1)
 - ▶ error (1)
- ▶ 용어정리 (4)



ARCHIVES

- ▶ April 2019 (2)
- ▶ March 2019 (1)
- ▶ February 2019 (1)
- ▶ January 2019 (15)
- ▶ December 2018 (9)
- ▶ November 2018 (20)
- ▶ October 2018 (6)
- ▶ September 2018 (8)
- ▶ August 2018 (26)
- ▶ July 2018 (13)
- ▶ June 2018 (1)

1 톰캣서버의 tomcat-users.xml 수정

젠킨스가 톰캣서버에 접근하여 빌드 결과물인 war를 배포하기 위해서는 톰캣에 접근할수 있는 계정이 필요하다.

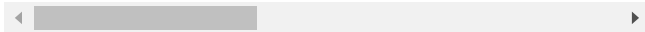
톰캣서버의 관리자 계정을 만들어서 이 계정을 젠킨스에게 알려주면 된다.

톰캣이 설치되어 있는 서버에 /etc/tomcat/tomcat-users.xml 파일에 아래 내용을 추가하자.

tomcat-users라는 루트 태그 내부에 추가하면 된다.

톰캣의 관리자 계정은 admin / admin 으로 설정했다.

```
1 <role rolename="manager-gui"/>
2 <role rolename="manager-script"/>
3 <role rolename="manager-jmx"/>
4 <role rolename="manager-status"/>
5 <role rolename="admin-gui"/>
6 <role rolename="admin-script"/>
7 <user username="admin" password="admin" roles
```



젠킨스가 사용할 톰캣 계정 설정

톰캣을 리스타트 하자.

```
1 systemctl restart tomcat
```

1 젠킨스에 svn 빌드+배포 잡 생성

▶ May 2018 (13)

▶ April 2018 (14)

▶ March 2018 (6)

▶ February 2018 (1)

▶ January 2018 (5)

▶ December 2017 (6)

▶ November 2017 (3)

▶ October 2017 (13)

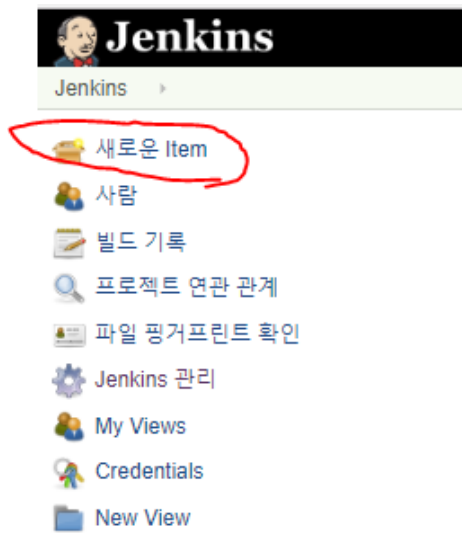
▶ September 2017 (18)

▶ August 2017 (10)

광고없는
플레이어
Vimeo에서 누리세요.

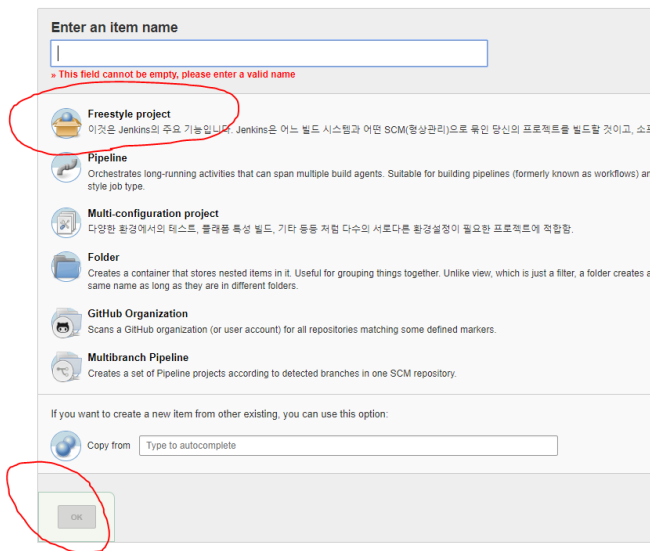
vimeo

자세히



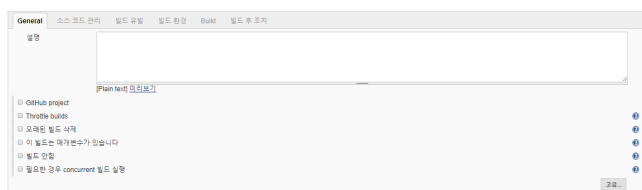
새로운 item 생성화면 이동

적절한 Job 이름을 입력하고, Freestyle project 를 클릭한뒤 ok를 클릭하자.



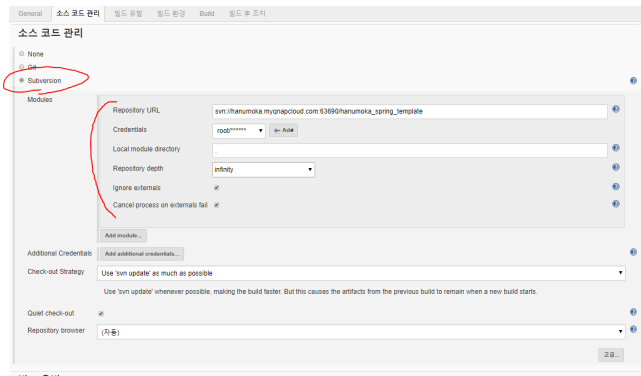
새로운 item(JOB)생성화면

그 다음에는 자세한 젠킨스 Job 설정화면이 탭으로 구별되어 나온다.



General 탭

General 탭에는 딱히 설정 할 것이 없다. 넘어가자.

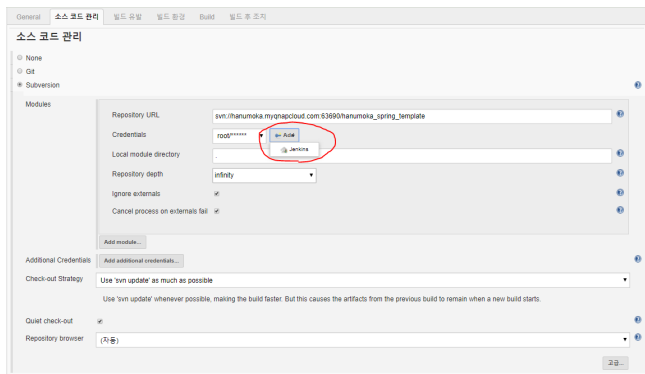


소스 코드 관리 탭

소스코드 관리 탭에서는 이 젠킨스 잡에서 빌드할 때 가져올 소스코드의 위치인 SVN 경로를 설정해야 한다.

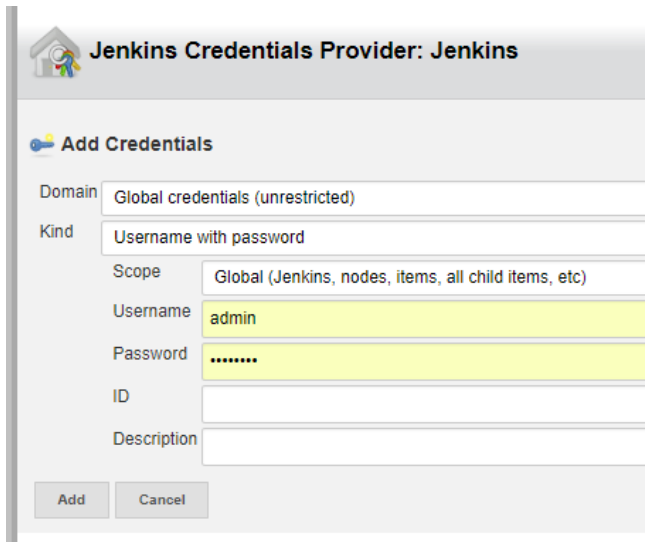
Subversion을 선택하고 SVN원격 리파지토리 URL과 리파지토리에 접근할수 있는 svn 계정정보를 입력하자.

svn계정은 read 권한만 있어도 된다.



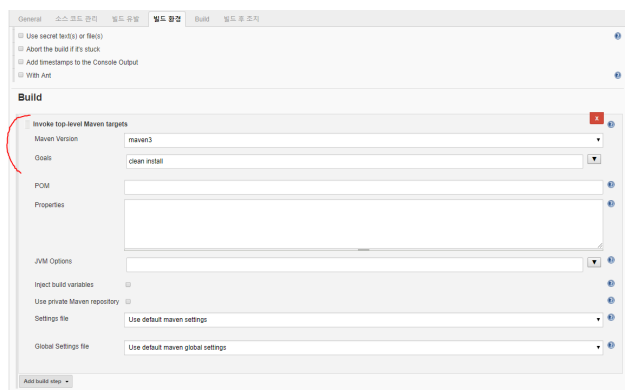
소스 코드 관리 탭 - svn 계정을 등록하려면 add버튼을 클릭해라.

Jenkins가 나오는데 그것을 클릭하면 아래처럼 젠킨스에서 계정정보를 관리하는 화면이 나온다. 젠킨스는 다양한 외부 시스템에 접속을 할 필요가 있는데, 그때 접속할 계정정보를 관리하는 화면이 이 화면이다.



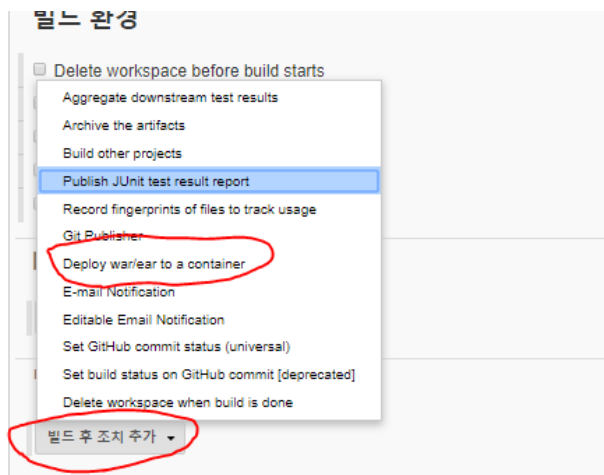
tomcat접속 계정정보를 입력하자.

입력을 끝내고 add를 누르면 소스 코드 관리 탭의 Credentials항목에서 입력한 계정을 선택 할 수 있다.



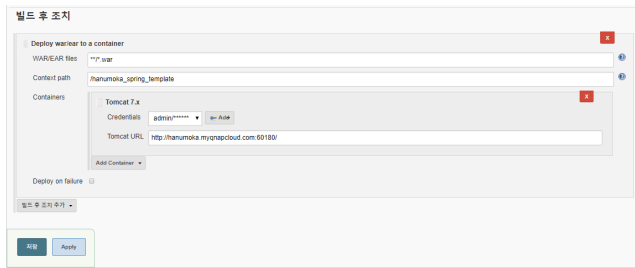
Build 탭으로 이동하자.

이글의 초반부에 설정한 maven을 선택하고 Goals에 maven 명령어인 clean install 을 입력하자.



빌드 후 조치 탭으로 이동하자.

빌드 후 조치 추가를 클릭하고 Deploy war/ear to a container 를 클릭하자.



빌드 후 조치에서 배포대상인 tomcat의 정보를 입력하자.

container항목을 선택하고 배포서버의 tomcat 버전인 tomcat7을 골라주자.

빌드 후 조치에서는 빌드 결과물인 war파일을 tomcat 서버에 배포하기 위한 설정을 해야 한다. 배포 대상인 war파일을 위 이미지 처럼 설정하고, context path를 설정하자.

이 context path는 war의 url경로가 된다.

그리고 아까 tomcat에 설정한 톱캣 관리자 정보를 입력하자.

Tomcat URL은 배포가 되는 Tomcat서버의 URL을 입력하면된다.

입력이 끝났다면 저장하자.

이제 모든 준비가 끝났다.

만든 젠킨스 잡을 실행하여 maven 웹 프로젝트를 빌드하고 톱캣에 배포해보자.

=====


jenkins 빌드



The screenshot shows the Jenkins web interface for the project 'hamumoka_spring_template_build_deploy'. The 'Build Now' button is highlighted with a red circle. The interface includes a sidebar with navigation links like '대시보드', '상태', '변경사항', '작업공간', 'Build Now', 'Project 설정', '구성', and 'Rename'. The main area shows the project name and a list of build jobs with their status and duration. A table at the bottom lists the build history with columns for job name, status, and time.

Job Name	Status	Time
Build	Success	2018. 5. 24 오후 3:44
Build	Success	2018. 5. 24 오후 2:40
Build	Success	2018. 5. 24 오후 2:02
Build	Success	2018. 5. 24 오후 1:40
Build	Success	2018. 5. 24 오후 1:40
Build	Success	2018. 5. 24 오전 11:28
Build	Success	2018. 5. 24 오전 10:51

[illegible]



Tomcat Web Application Manager

Message: or

Manager
[List Applications](#)

[HTML Manager Help](#)

[Manager Help](#)


Applications

Path	Version	Display Name	ReadOnly	Senders	Commands
/	<i>None specified</i>		True	2	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/docs	<i>None specified</i>	Tomcat Documentation	True	2	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples	<i>None specified</i>	JSP and JSP Examples	True	2	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples/WEB-INF/jsp	<i>None specified</i>		True	2	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes

10/15

← → ↺ 192.168.0.122:8080/hanumoka_spring_template/

Hello world! 한글입력 테스트

 대리님의 복귀

디데이까지 35일 14시간 17분 17초 남았습니다.



The time on the server is 2018년 5월 26일 (토) 오후 6시 42분 27초.

배포 war 동작 확인

배포된 내용이 반영되어 동작하는 것을 확인 할 수 있다.

추가 : 빌드중 오류 발생

이번 실습에서 빌드중 발생한 오류내용을 추가한다.

젠킨스 잡에서 maven 빌드시 아래 오류메세지들이 발생했다.

오류 내용 :error: unmappable character for encoding UTF8

찾아보니 윈도우 환경에서 개발된 소스가 리눅스 환경에 빌드될때 발생하는 인코딩 문제라고 한다.

```
[INFO] -----
[ERROR] COMPILATION ERROR :
[INFO] -----
[ERROR] bootstrap class path not set in conjunction with -source 1.6
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,33] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,34] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,35] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,36] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,37] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,39] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,41] error: unmappable character for encoding UTF8
[ERROR]
/var/lib/jenkins/workspace/hanumoka_spring_template_build_deploy/src/main/java/com/hanumoka/web/HomeController.java:[49,42] error: unmappable character for encoding UTF8
```

maven빌드 시 error: unmappable character for encoding UTF8

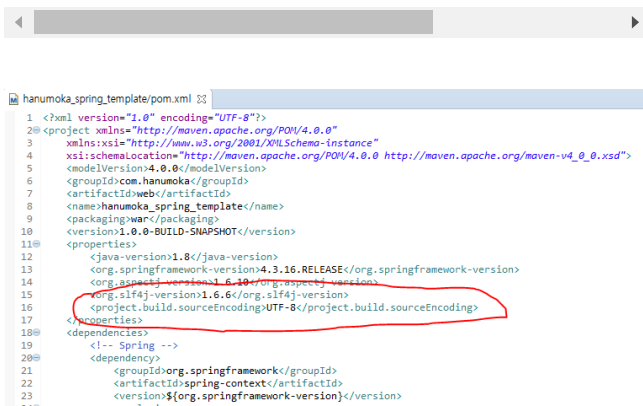
개발환경의 스프링 maven 프로젝트의 pom.xml에 UTF-8 인코딩 설정을 추가해야한

다.

pom.xml 아래 내용을 추가한다.

일단 properties 항목에 아래 내용을 추가하자.
maven이 빌드할때 아래 프로퍼티를 참조하여 인코딩을 정한다고 한다.

```
1 <project.build.sourceEncoding>UTF-8</project.
```



Maven빌드 UTF-8 인코딩 설정

그리고 maven-compiler-plugin 플러그인의 configuration에 아래 내용을 추가한다.

```
1 <encoding>UTF-8</encoding>
```

```
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.5.1</version>
  <configuration>
    <source>1.6</source>
    <target>1.6</target>
    <encoding>UTF-8</encoding>
    <compilerArgument>-Xlint:all</compilerArgument>
    <showWarnings>true</showWarnings>
    <showDeprecation>true</showDeprecation>
  </configuration>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  ...
```

maven-compiler-plugin UTF-8 인코딩 설정

로컬에서 maven 빌드를 해보면 이제는 로컬에서
error: unmappable character for encoding
UTF8 오류가 발생하는 것을 확인 할 수 있다.

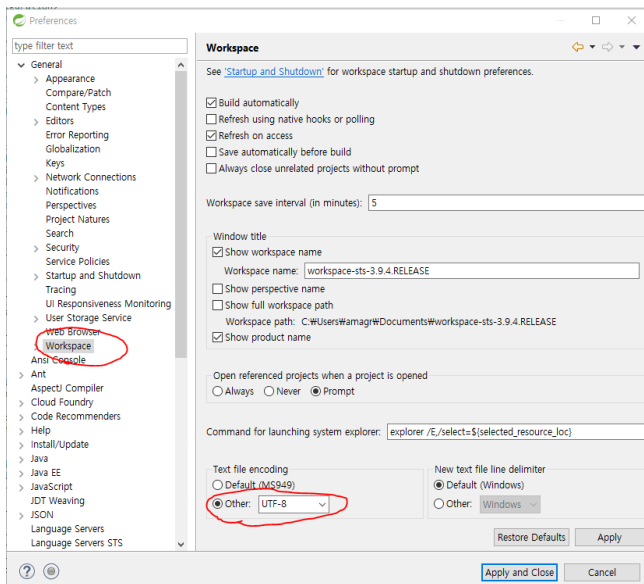
```

le (default-compile) on project web: Compilation failure: Compilation failure:
ain\java\com\hanumoka\web\HomeController.java:[49,33] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,34] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,35] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,36] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,37] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,39] error: unmappable character for encoding UTF-8
ain\java\com\hanumoka\web\HomeController.java:[49,41] error: unmappable character for encoding UTF-8

```

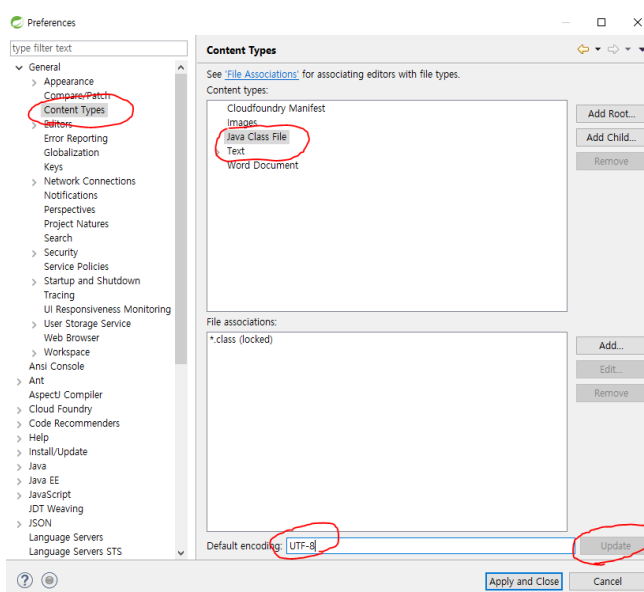
STS에서 maven 빌드시 error: unmappable
character for encoding UTF8d 발생

개발도구인 STS(Spring Tool Suite) 또는 이클립스에서도 UTF-8 인코딩 설정이 필요하다.



Window - Preferences - General - Workspace 에
서 Text file encoding에 UTF-8 설정을 하자.

그리고



Window - Preferences - General - Content Types
에도 UTF-8을 설정하자

Java Class File과 Text에 Default encoding을 UTF-8을 입력하고 update를 클릭하여 각각 인코딩을 적용하자.
주의할 점은 개발툴 UTF-8인코딩 설정을 하면 자바 파일등에서 기존에 작성한 한글등이 깨진다.

그 뒤 로컬에서 maven 빌드시 빌드 성공 확인!!!

이제 수정된 내용을 SVN에 commit하고 젠킨스에서 다시 빌드+배포를 돌리자.

젠킨스에서 오류없이 빌드+배포 되었으며 tomcat에서 결과를 확인 했다.

끝!!!

Share |

댓글 커뮤니티



♡ 추천

🐦 Tweet

f 공유

인기순 ▼

토론 참여하기



임응식 • 24일 전

svn에서 소스 다운로드 한뒤 war파일로 빌드되기전에 별도 작업을 수행해야하는데 이런 경우 어떻게 해야하나요?
(pre build라는게 없어서ㅜㅜ)
^ | v • 답글 • 공유



woosang Jeon • 2달 전

안녕하세요 작성하신글 잘읽었습니다.
궁금한점이 있는데 작성글 초반에 '윈도우 pc 1개, centos7 3개' 이렇게 필요하다고 되어 있는데, 메이븐, 톰캣, jdk 는 모두 1대의 pc 에 모두 설치한건가요?
그럼 윈도우pc 와 리눅스 pc 각각1 대가 필요한건가요???



© 2019 Hanumoka

Powered by Hexo. Theme by PPOffice