**1강의**

<http://bitly.com/1lv81DY> 접속해서 강의 자료를 다운 받습니다.

개발환경

* Eclipse
* SVN 플러그인 설치
* SVN Connecter 플러그인 설치
* 빌드도구 Maven , Gradle , Ant => 강좌는 Maven으로 진행 합니다.

빌드툴

Nexus (Maven Remote Repository)

CI툴

Jenkins

시작

해당 파일 디렉토리

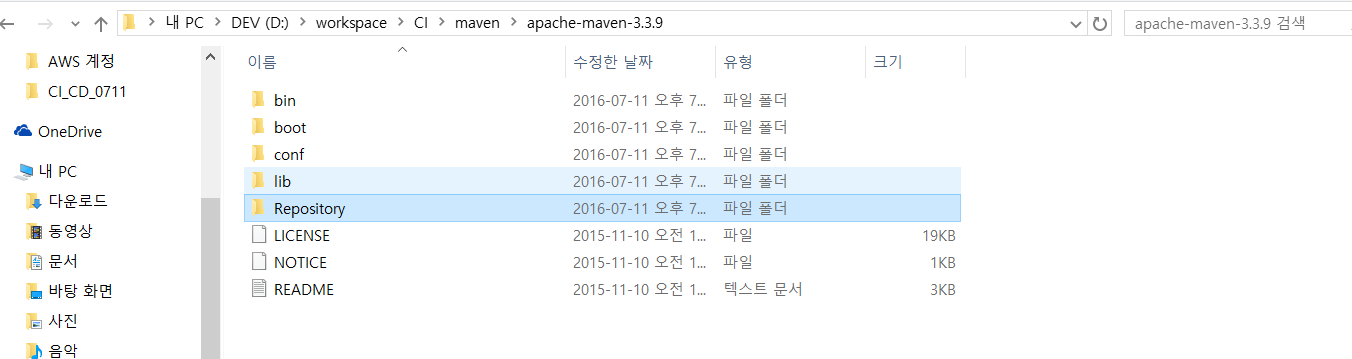
D:

1. Maven DownLoad

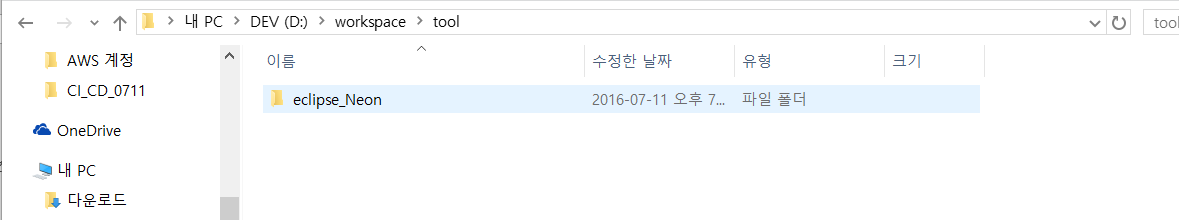
<http://maven.apache.org/download.cgi>

다운로드 대상파일 : [apache-maven-3.3.9-bin.zip](http://apache.tt.co.kr/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.zip)

압축풀기

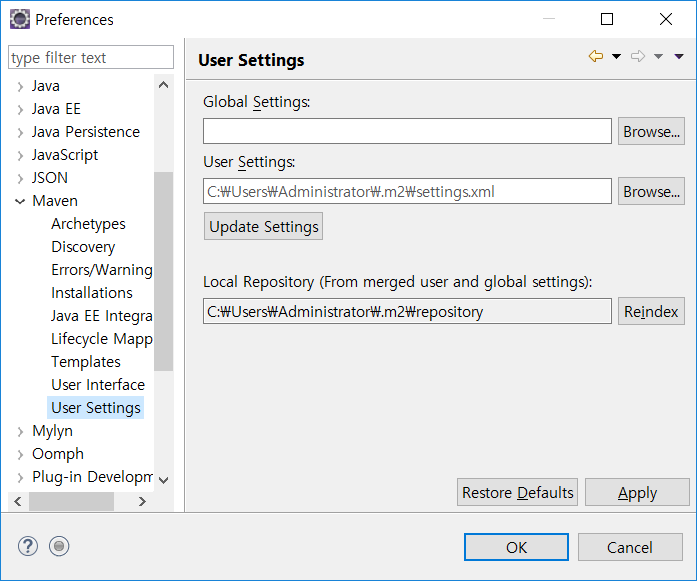


1. [www.eclipse.org](http://www.eclipse.org) 이클립스 다운로드 네오가 최신버전이고 JDK 1.8 버전이 필요로함

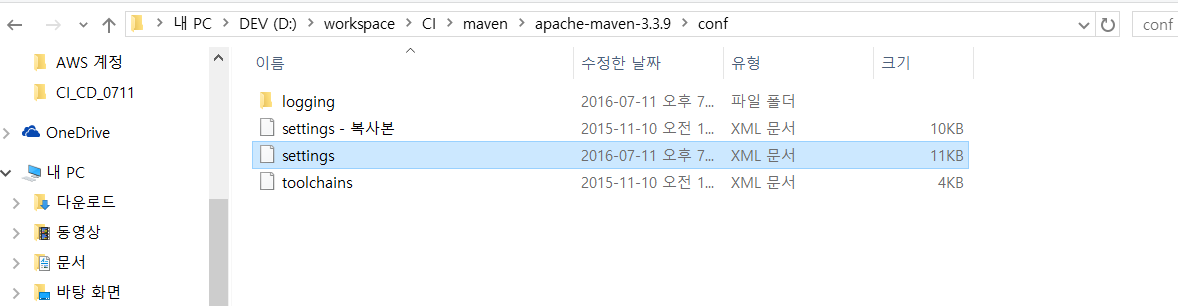


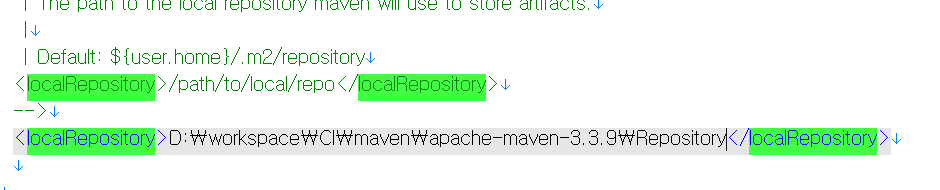
1. Maven Repository 설정

디폴트

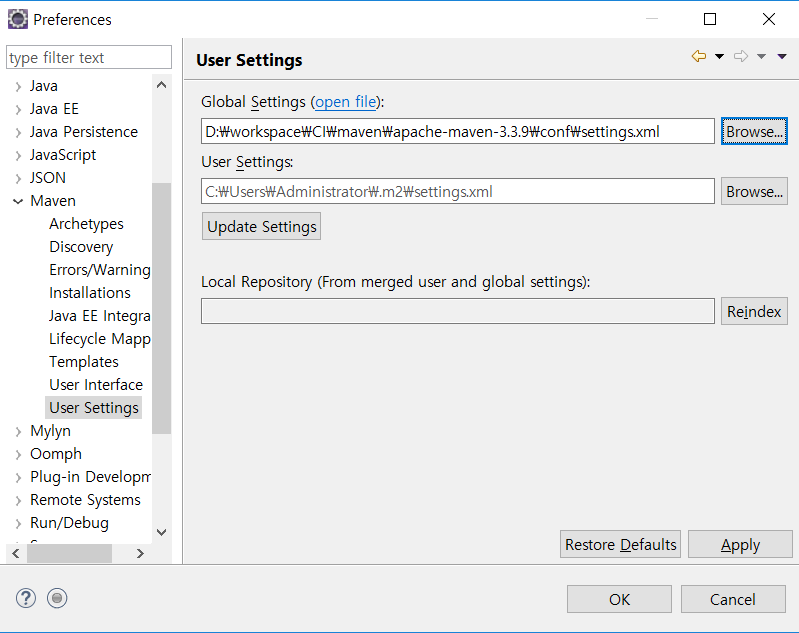


변경

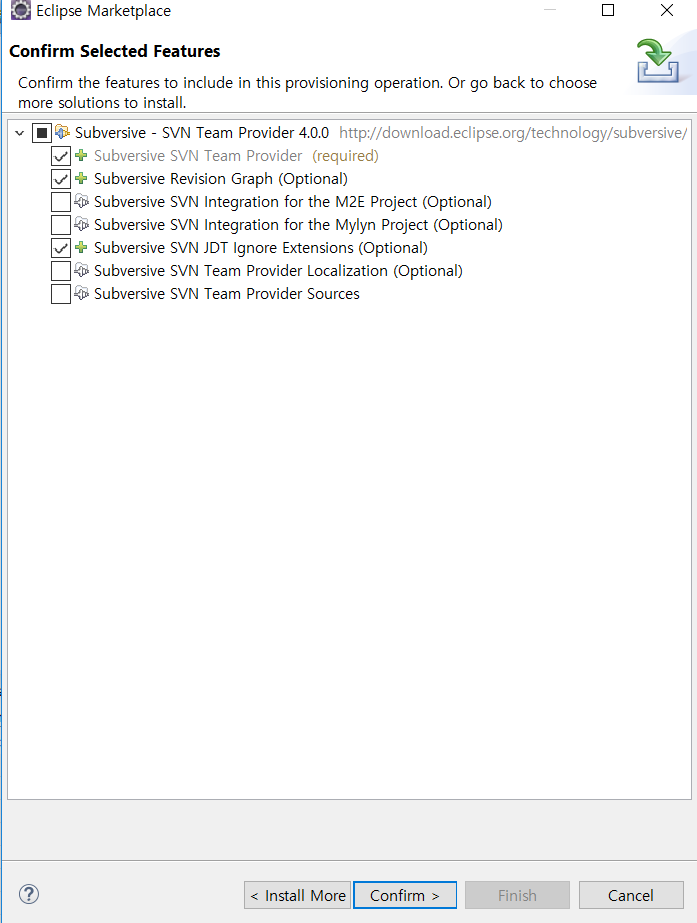


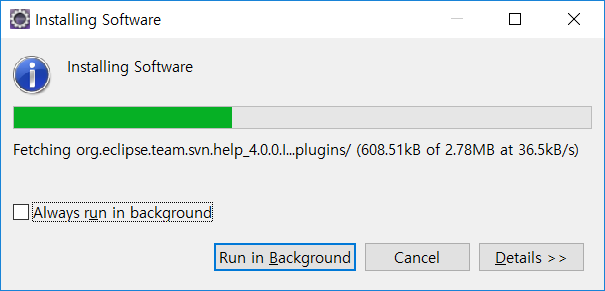


위와 같이 변경 후 적용



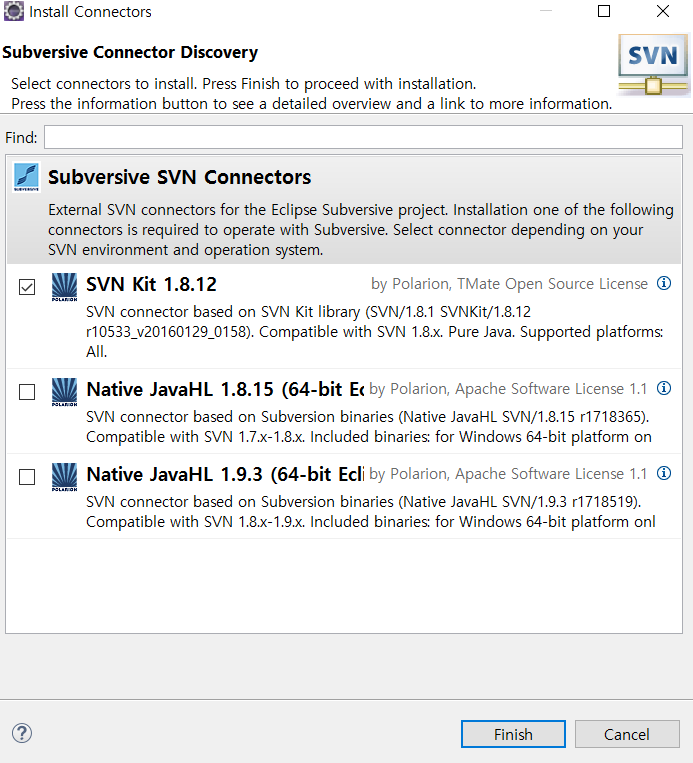
SVN Eclipse에서 추가





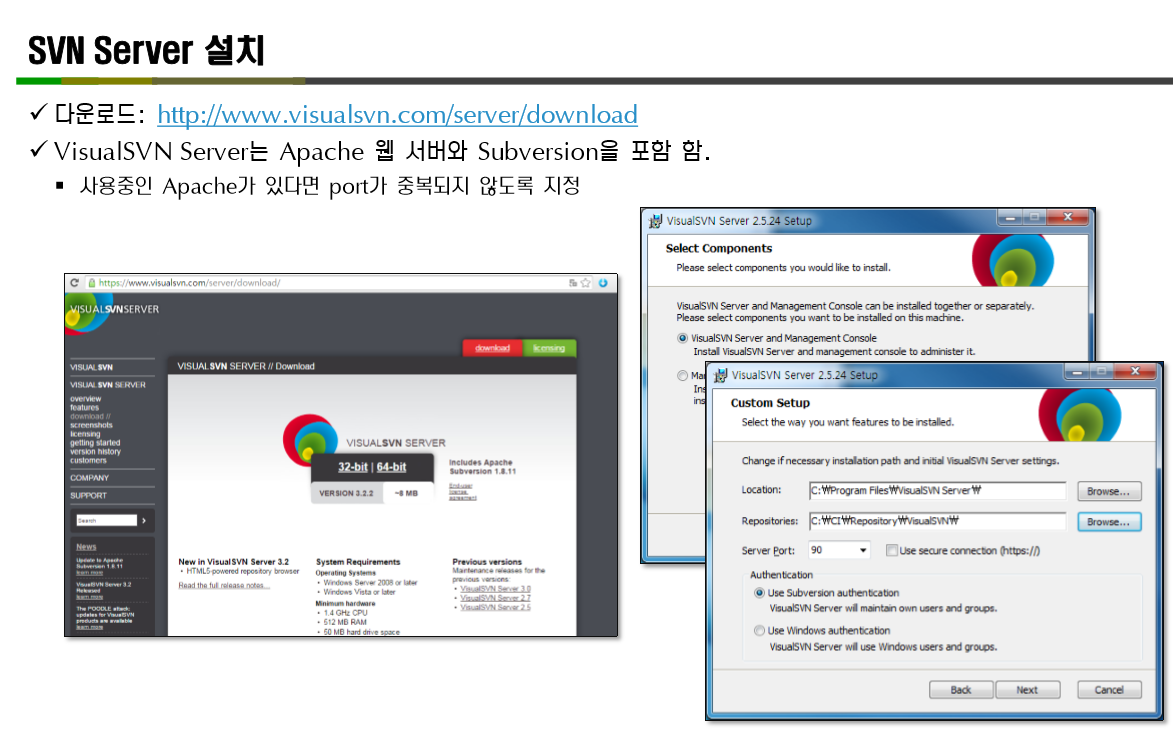
설치 완료 후 이클립스 restart 합니다.

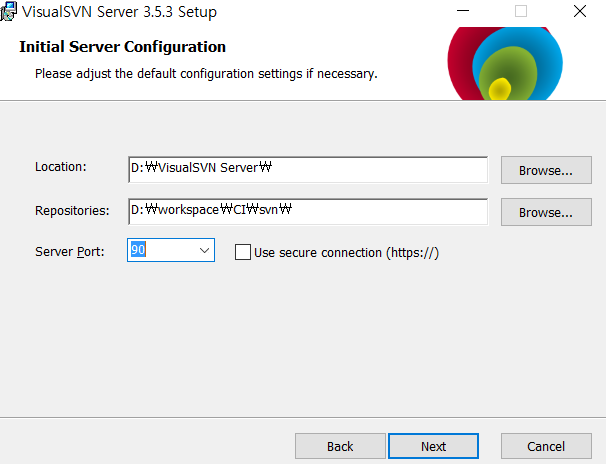
**Window > Preferences > team > SVN 선택**



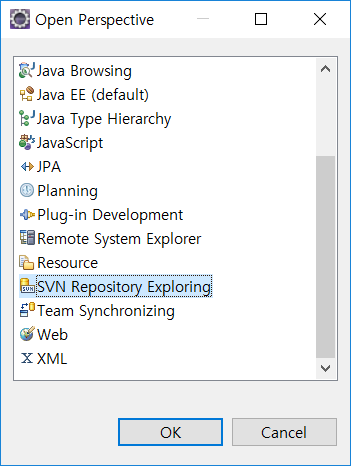
SVN Server 설치

[www.visualsvn.com/server/download](http://www.visualsvn.com/server/download) 에서 64bit 다운받습니다.

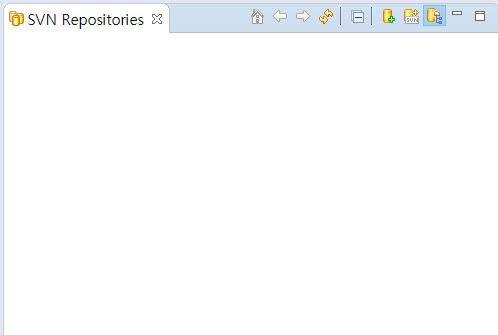


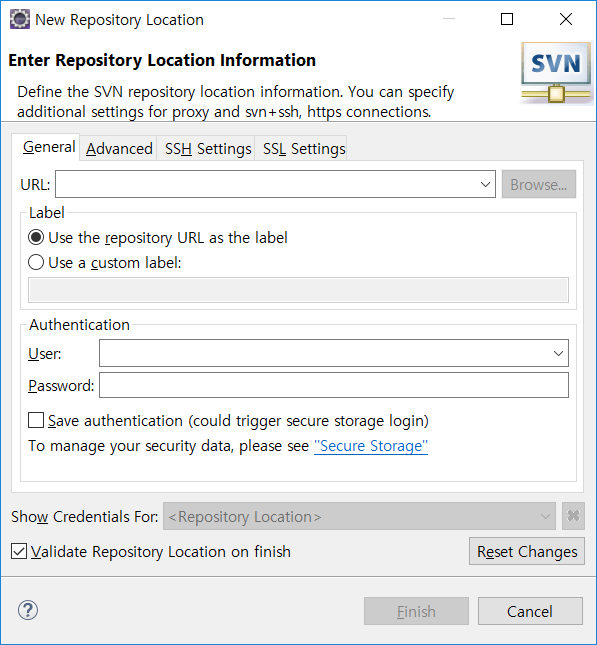


다시 이클립스로

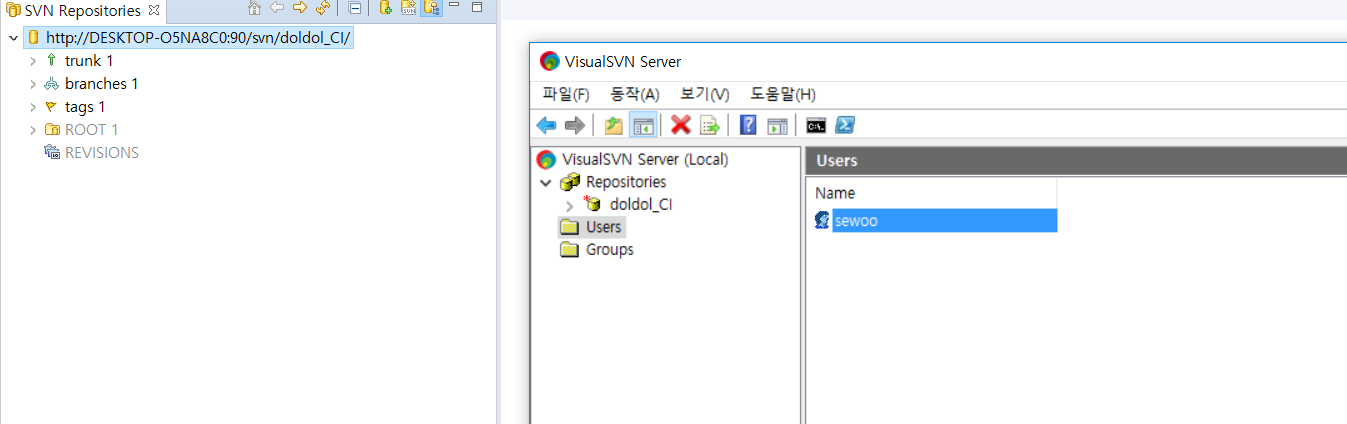


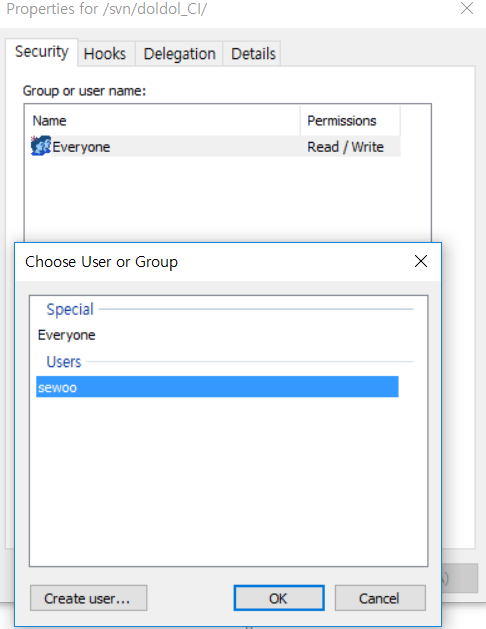
선택하면 아래 SVN Repositoryes 생성됨

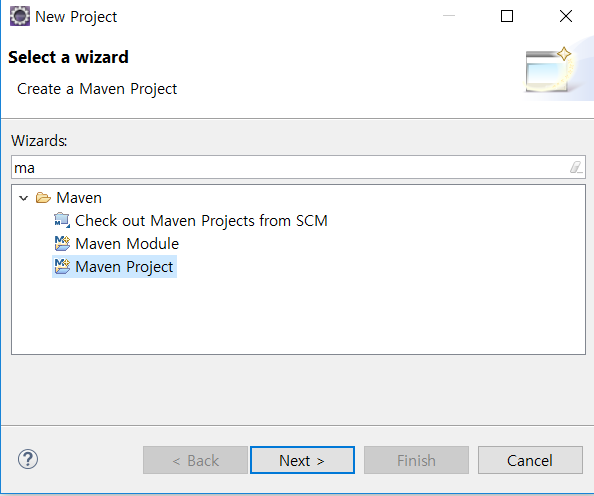


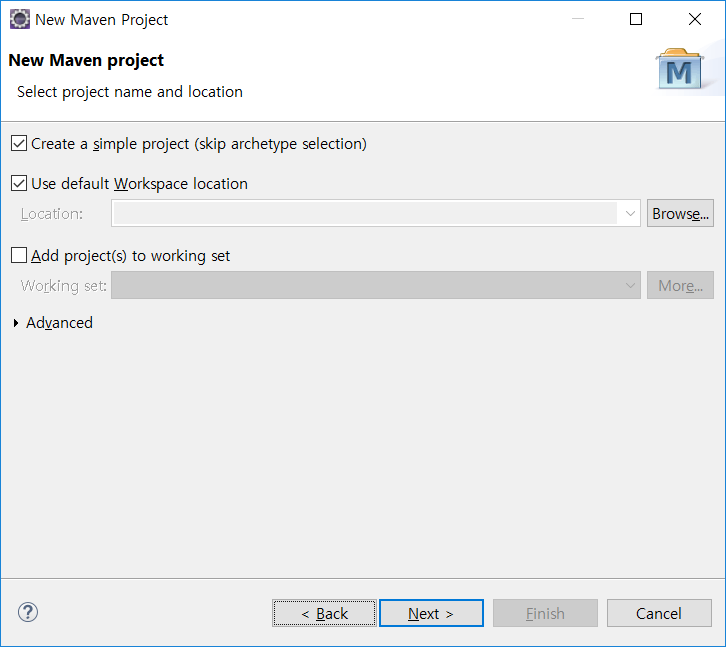


아까전에 만들었던 접속이 되는 부분을 확인 하였습니다.

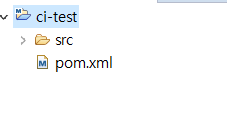


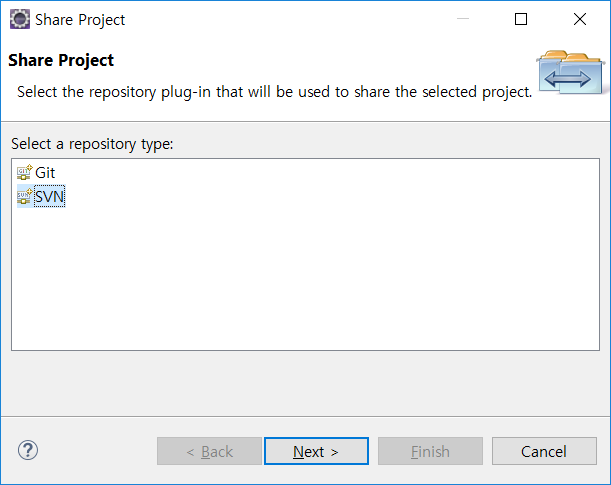


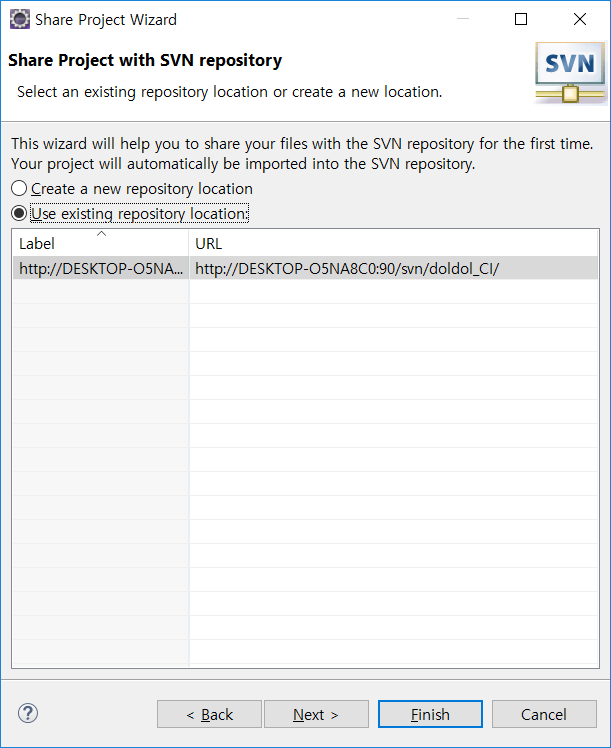




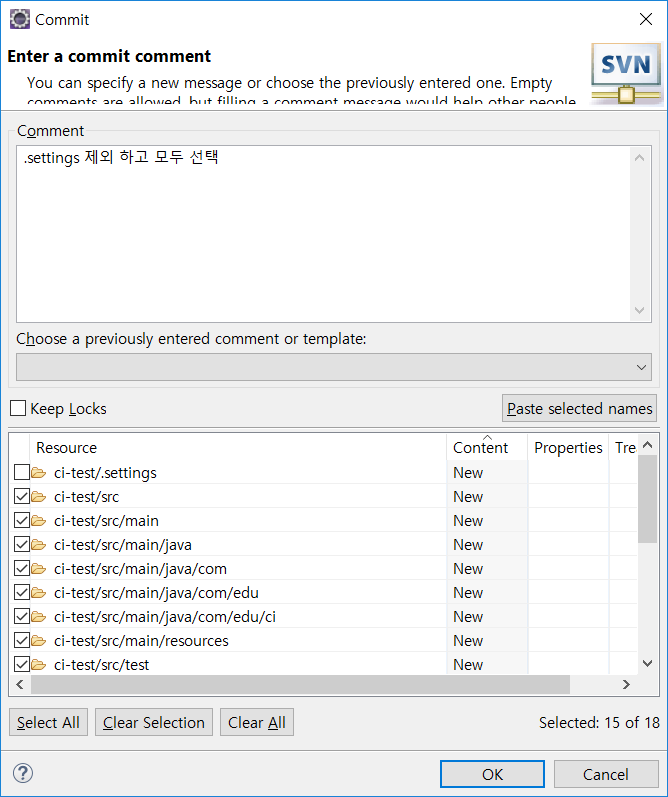
생성된 프로젝트



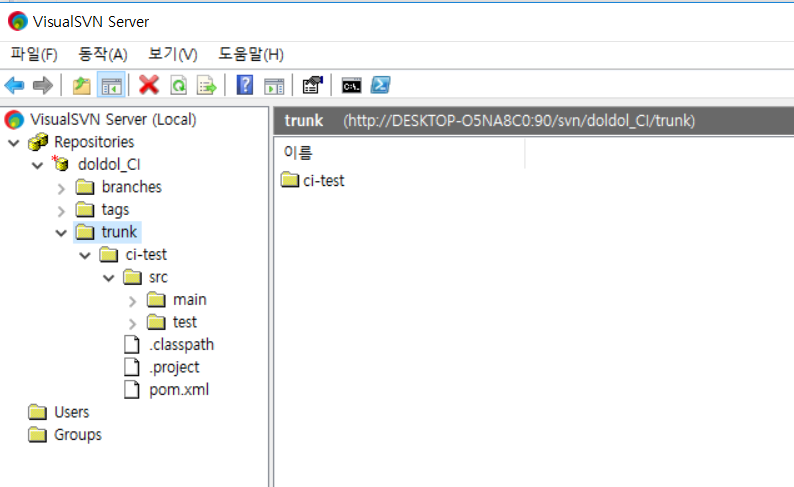


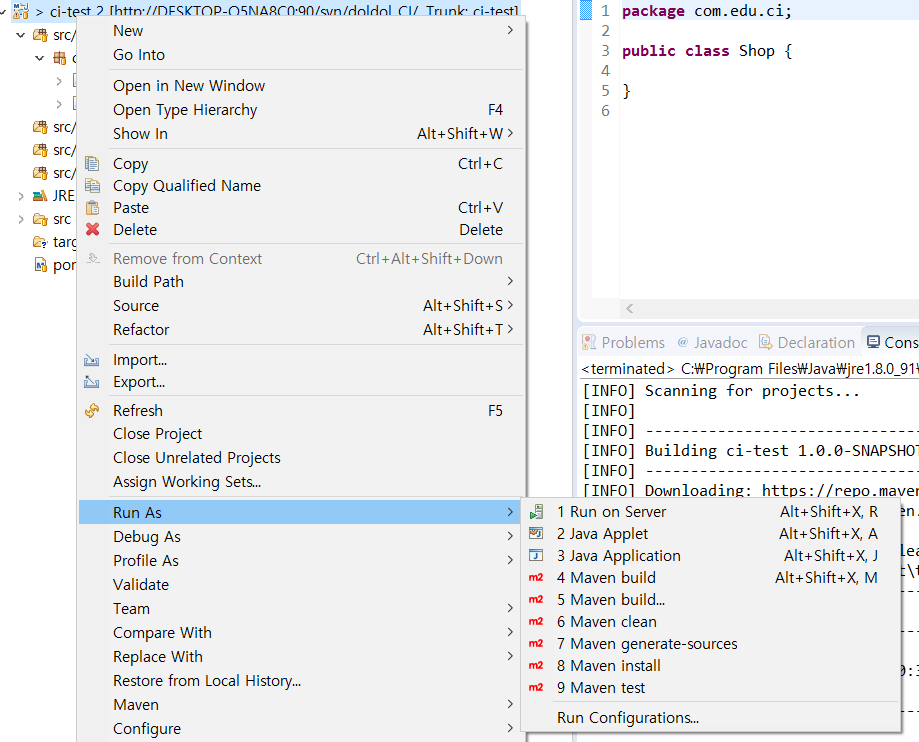




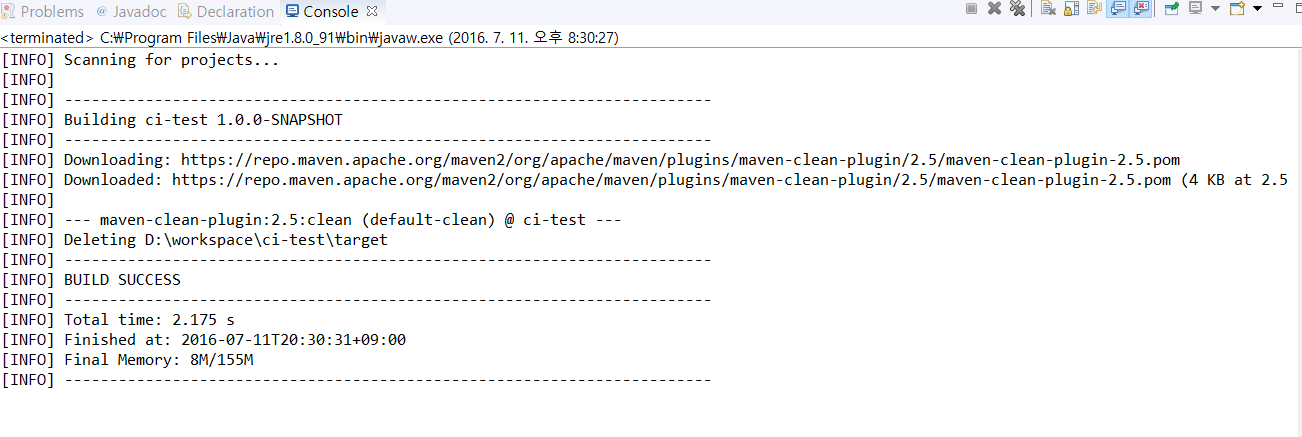


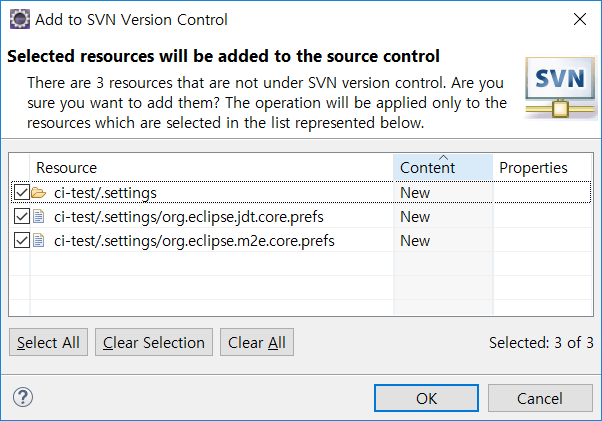
Commit 하고 서버 확인



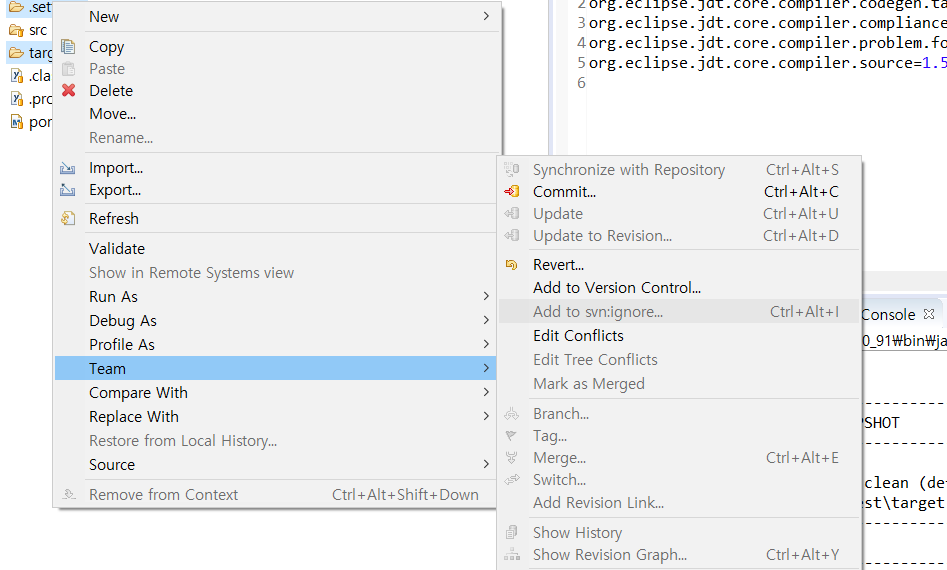


Maven Clean

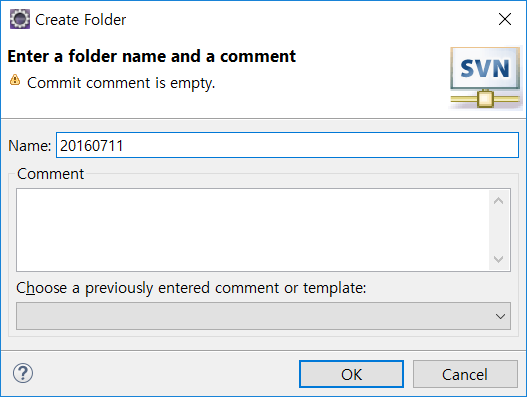


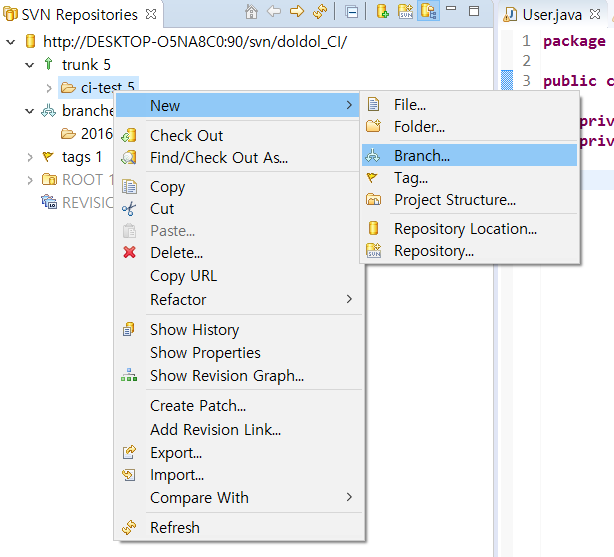


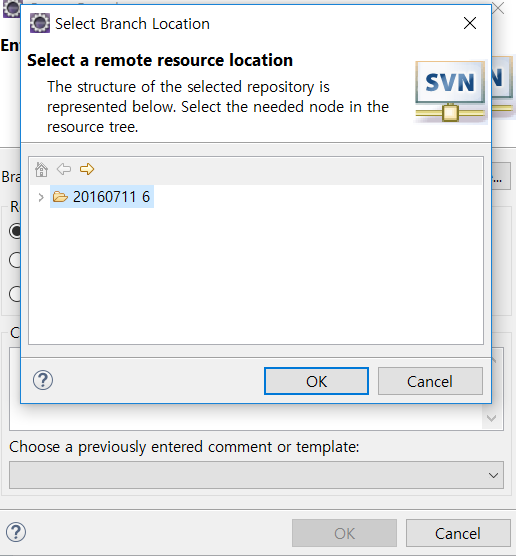
필요없는 파일 ADD to svn:ignore… 선택하면 커밋 대상에서제외



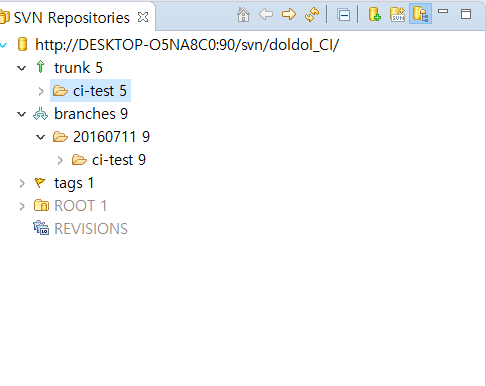
Branches 만들어 보자



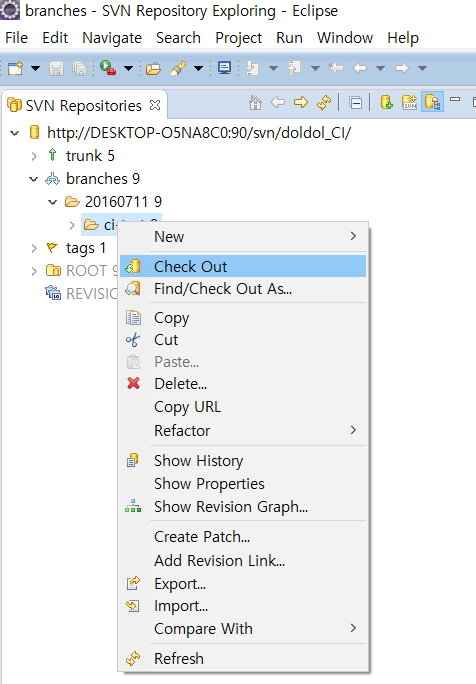


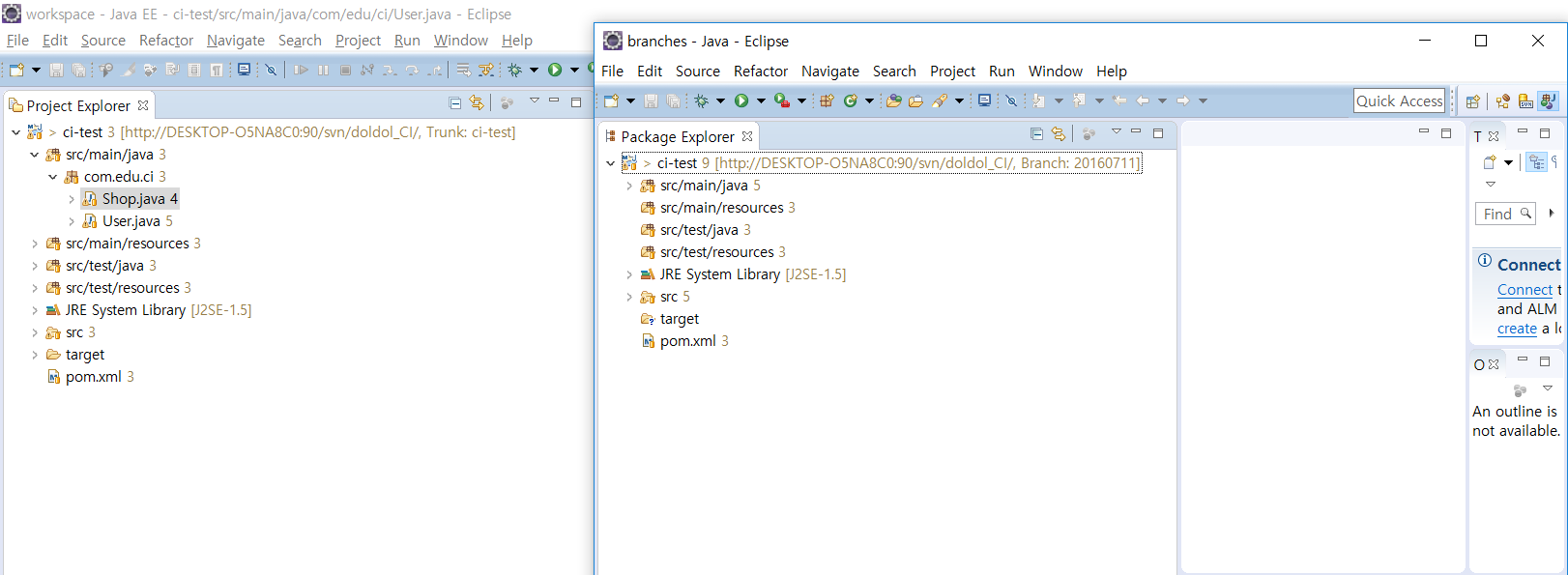


Branches 에 동일하게 생성됨

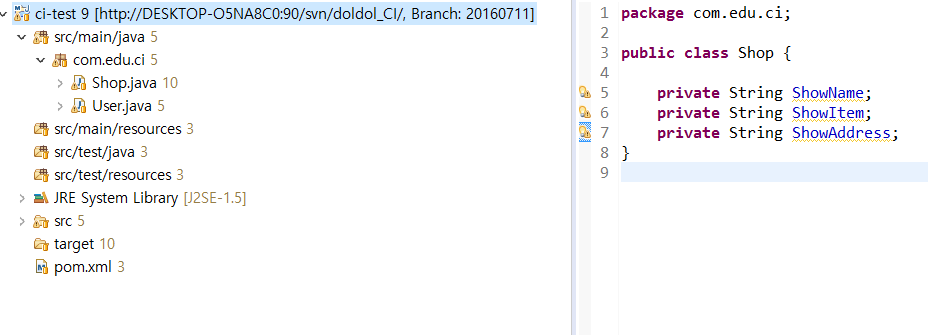


신규 이클립스를 실행 해서 새로 만든 branches에서 checkout 합니다.

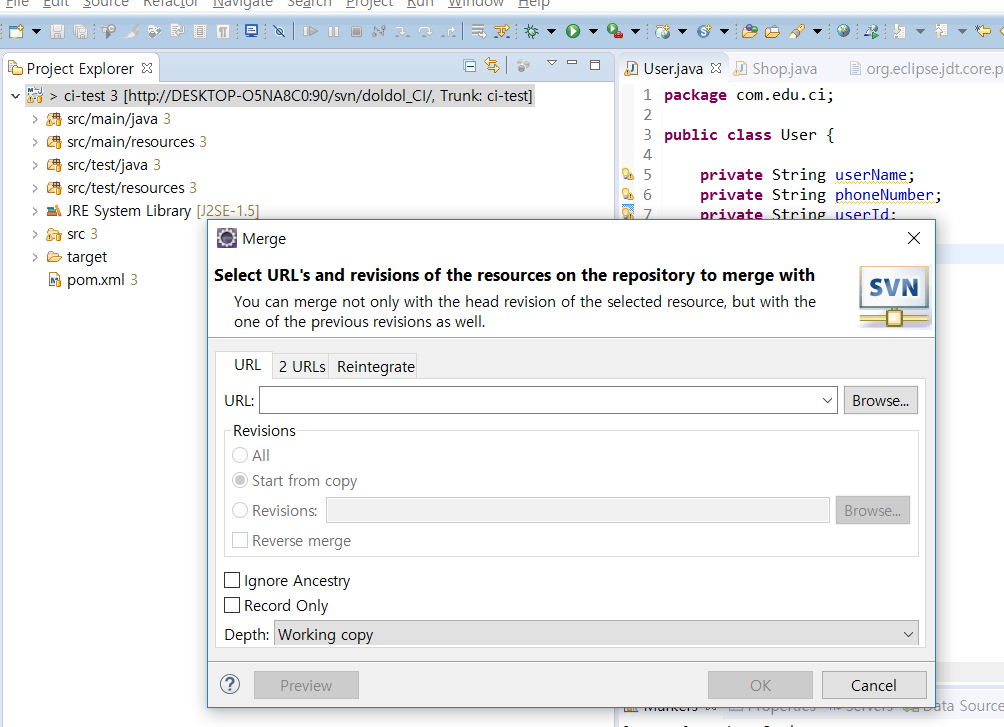


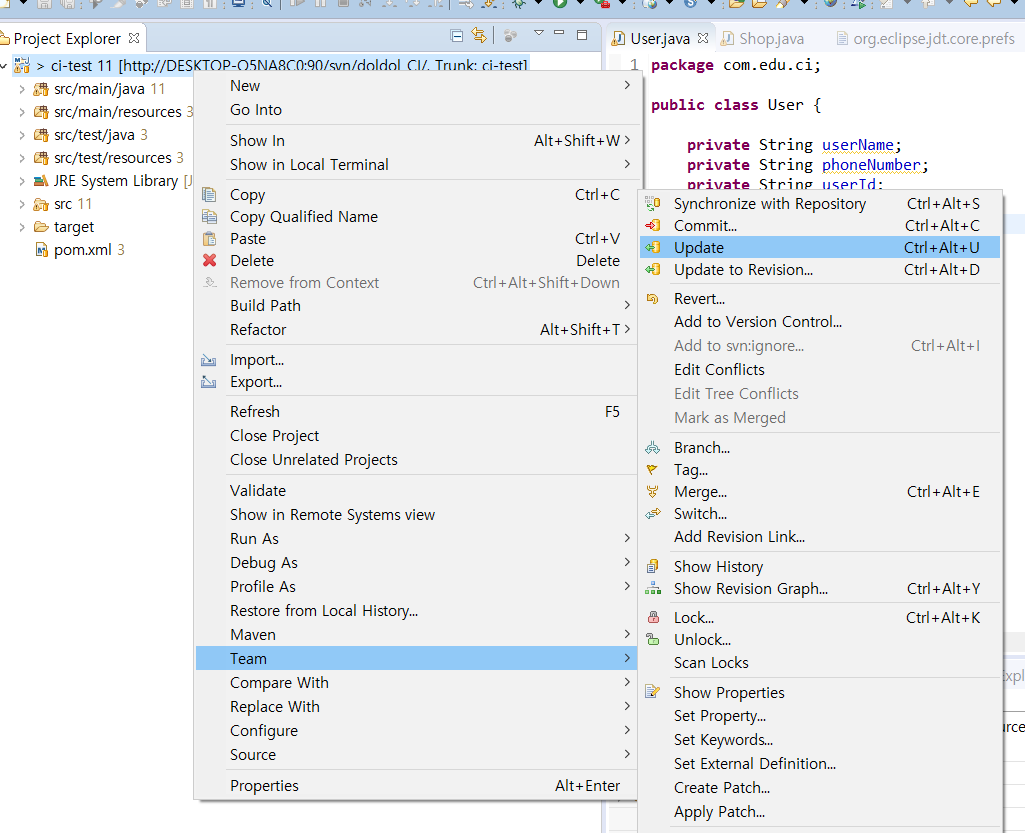


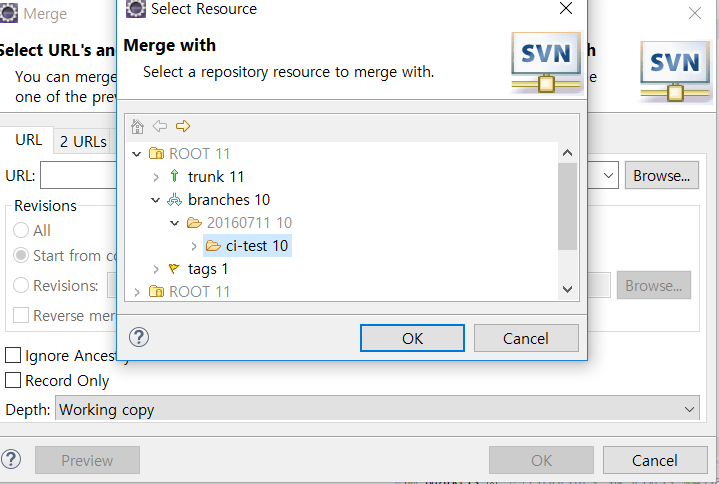
Branch에 필드 하나 추가하고 Show commit 합니다.

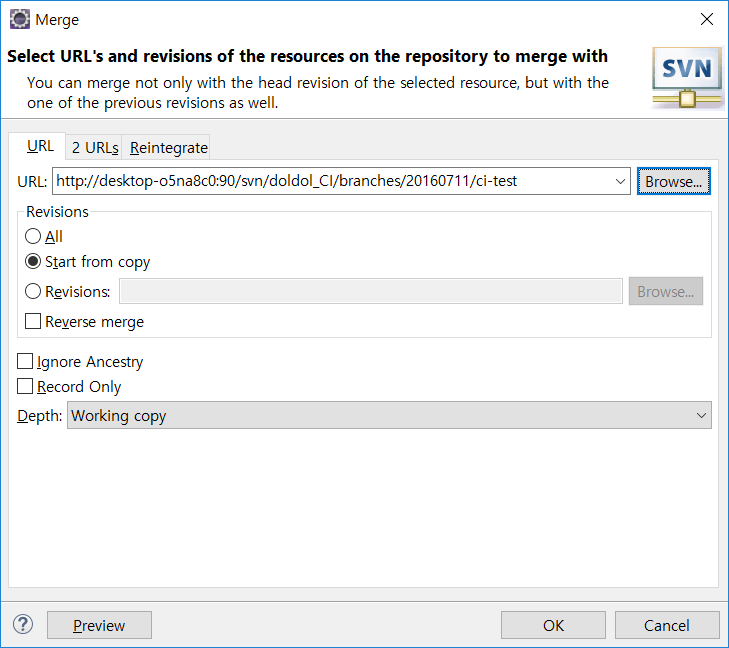


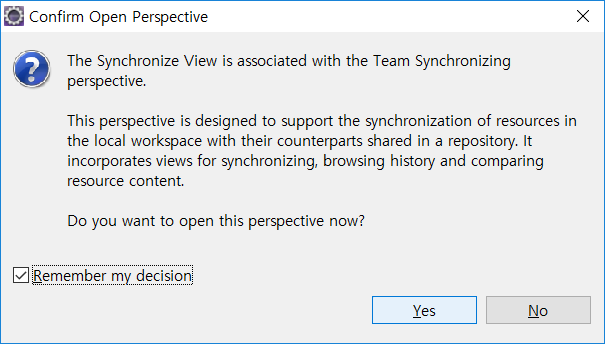
Trunck 에 User 클레스에 필드 하나 추가하고 Commit 합니다.





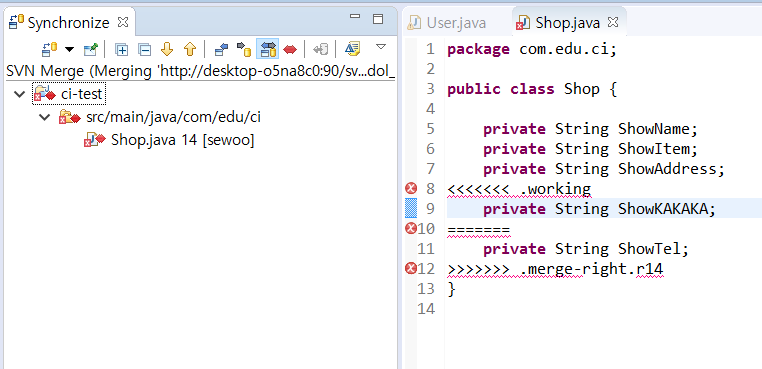




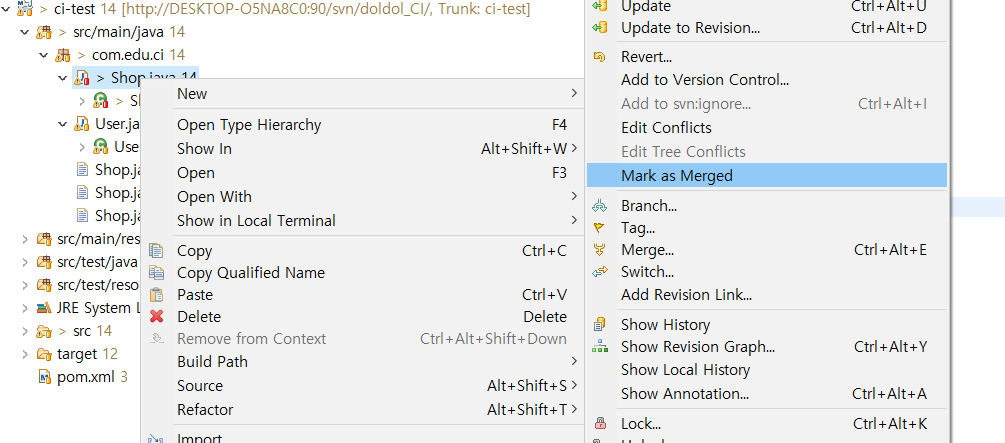


trunk에서 Shop 바꾸면 충돌 당연히 branch 에서도 Shop 클레스 수정이 되어 있음

충돌 테스트



이럴 때 Mark as Merged 를 사용하여 동기화 시킨다



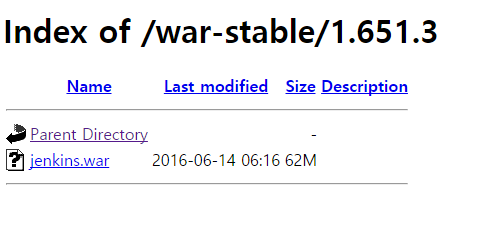
CI – 설치 Jenkins

다운로드 URl <http://jenkins-ci.org>

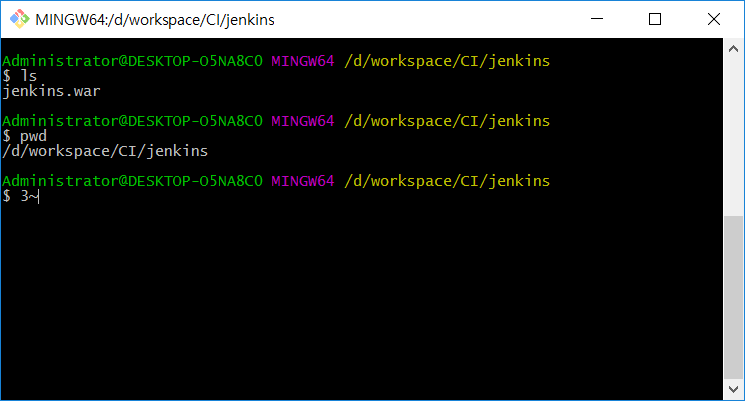


<http://mirrors.jenkins-ci.org/war-stable/>

<http://mirrors.jenkins-ci.org/war-stable/1.651.3/>



다운받은 경로로 이동합니다.



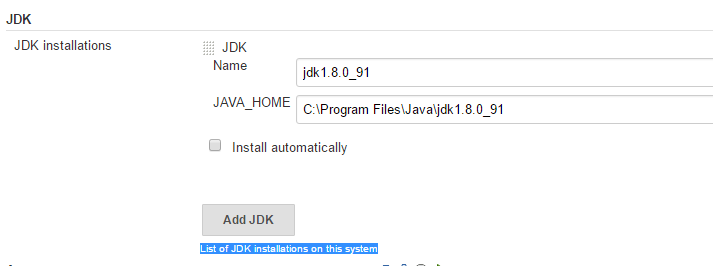
실행합니다.



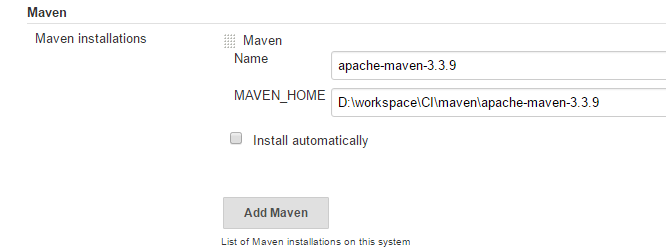
실행된 화면



Jenkins 관리 > 시스템 설정 > JDK add



Jenkins 관리 > 시스템 설정 > Maven add



# Jenkins 관리 > Configure Global Security

# 

# 계정생성

# 

# 

# 비밀번호doldol44

# 계정을 생성후 다시

# Jenkins 관리 > Configure Global Security > Matrix-based security

# 

# 다시 이클립스 Run-AS

# 

# Jenkins새로운 item

# 

**2강의**

# 늦어서 다노침 ㅜㅜ

# 

# 

# 위의 내용중 this link를 클릭 합니다.

# 

# Nexus 설치

# <http://www.sonatype.com/download-oss-sonatype>

# 

# 위의 내용중 zip 파일을 다운로드 받습니다.

# 압축푼 곳에서 실행 디렉토리로 이동

# 

# Console-nexus 실행

# 

# 설정파일

# 

# URL <http://localhost:8081/nexus/#welcome> 접속

# 

# 로그인

# Id - admin

# PW - admin123

# 

# 로그인 성공 화면

# 

# 호스티드 2개 생성

# 

# 

# 

# 그룹생성

# 

# 

# 권한생성 privlieges

# 

# 이미있는거 검색

# 

# 

# rule생성

# 

# 

# 

# 어디민롤

# 

# user생성

# 

# 

# doldol44

# doldol44

# 어드민도 동일하게 추가합니다.

# 

# 추가한 결과

# 

# Nexus 연동 설정 maven setting.xml

# 

# 

# 

# 위의 주소를 아래에 붙여 넣기 합니다.

# 

# 

# 아무 JAR 파일 복사

# 

# 

# 

# Select Artifact to upload 에서 jar 파일 선택

# 

# Add Artifact

# 

# Upload Artifact

# 

# 

# 

# 위의 설정을 POM에 작성

# 

# 

# Run 하면 에러남. 어디에 업로드 해줄지 설정해야 합니다. Pom.xml

# 아래내용 추가

# 해당 주소는

# 

# 

# 

# 다시 Run-as의 빌드 실행

# 

# 아래 성공 nexus 메시지 표시

# 

# 젠키스로 디플로이 해보자

# 

# <http://DESKTOP-O5NA8C0:90/svn/doldol_CI/branches/20160711/ci-test>

# 

# 

# 저장 후 실행

# 

# 

# 

# 3강의

# 여러 개의 WAR 혹은 JAR 생성

# 의존관계 POM 만 Jenkins 에 연결하면 됩니다.

# POM [ root ] 최상위 [ Gradle 의 멀티프로젝트 ]

# L WAR 프로젝트 1개 ( JSP / JS / controller )

# L JAR1 – Process (interface / logic) + L JAR3 Process (결제 Logic)

# L JAR2 – DB

# 위와 같이 여러 개의 프로젝트 구성을 해봅시다.

# 간단하게

# POM 1개

# JAR 2개 정도 만들어서 의존관계 Deploy를 해봅시다.

# Maven 프로젝트 생성

# 

# 

# 

# 기존에 만든 프로젝트 때문에 ci-basci1🡨 1 붙인거 입니다. 1 빼야합니다.

# 동일 하게 pom이 아닌 jar 2개를 만듭니다.

# 

# Parent Project 에서 부모 ci-basic 를 작성해 줘야 합니다.

# 

# 

# Pom 프로젝트의 pom.xml에 위의 내용을 추가 합니다.

# 그리고 그 외 디렉토리는 삭제

# 프로젝트 SVN Commit 경로 설정

# ci-basic

# 

# 

# 

# trunck/ci-basic 해당 경로를 꼭꼭 확인해보자

# 두번째 ci-entity svn에 넣어보자 경로를 잘 확인해야 한다.

# 

# 

# Finish를 선택하면

# 

# 여기서 .setting .project 관련된건 모두 빼자

# 그후 pom.xml 수정 후 다시 commit

# 

# ci-process도 동일하게 진행한다.

# 

# 

# 

# 

# 넥스트를 선택하면 좀더 디테일한 경로 확인

# 

# 위의 내용 뺄거 빼고 Ok 선택후 진행한다

# 이클립스 디렉토리 구조

# 

# visualSVN Server도 확인해보자

# 

# Nexus 에 배포설정을 하기위해 POM ci-basic 수정하자

# 

# 위의 kosta-snapshots 내용을 추가 한다.

# Pom.xml를 commit

# JENKINS로 배포해보자

# 

# 

# 

# 배포하자

# 

# Jenkins 관리 > 플러그인관리

# 

# 재시작없이 설치

# 

# 

# 

# Maven setting.xml 파일 변경

# 

# ---- setting.xml 변경후 commit

# Jenkins 다시 배포 아까 플러그인 설치한걸 이용해봅니다.

# 

# 플러그인 설치 후 신규 생성

# 

# release:prepare - 먼저 실행 1.0.0 -> 1.0.1 전달

# release:perform - 체크아웃 빌드 -> Jenkins 로컬 떨구고 -> nexus 던짐

# 

# 위의 clean deploy 안넣는 부분 확인

# 

# 

# 빌드실행

# 클릭

# 

# 

# 기존 릴리즈 폴더 삭제

# 

# 

# 기존에 저렇게 들어 있엇음

# 

# 

# 

# 빌드 프로젝트 하나 복사해서

# 

# 맨앞에가 잘 빌드되면 뒤에꺼 실행

# 4강의

# 

# 

# 

# 기본 메이븐 디렉토리 생성

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# ci-web pom.xml 에서 수정해야 합니다.

# 

# 

# 

# 

# Jenkins

# 자식 프로세서 죽이지 않을려면

# <https://wiki.jenkins-ci.org/display/JENKINS/ProcessTreeKiller>

# 원격 배포시

# 

# 

# 테스트 스킵 하고 빌드

# 

# 해당 빌드 프로젝트만 setting.xml 참고하여 install 합니다.

# 플러그인 emma

# 

# 

# Emma plugin 테스트케이스를 적용 안하면 빌드 안됨

# 

# Golals and option [ Clean emma:emma install ]