

Eclipse – Git 연동

광명융합기술교육원
데이터분석과

기존 프로젝트 gitHub에 올리기

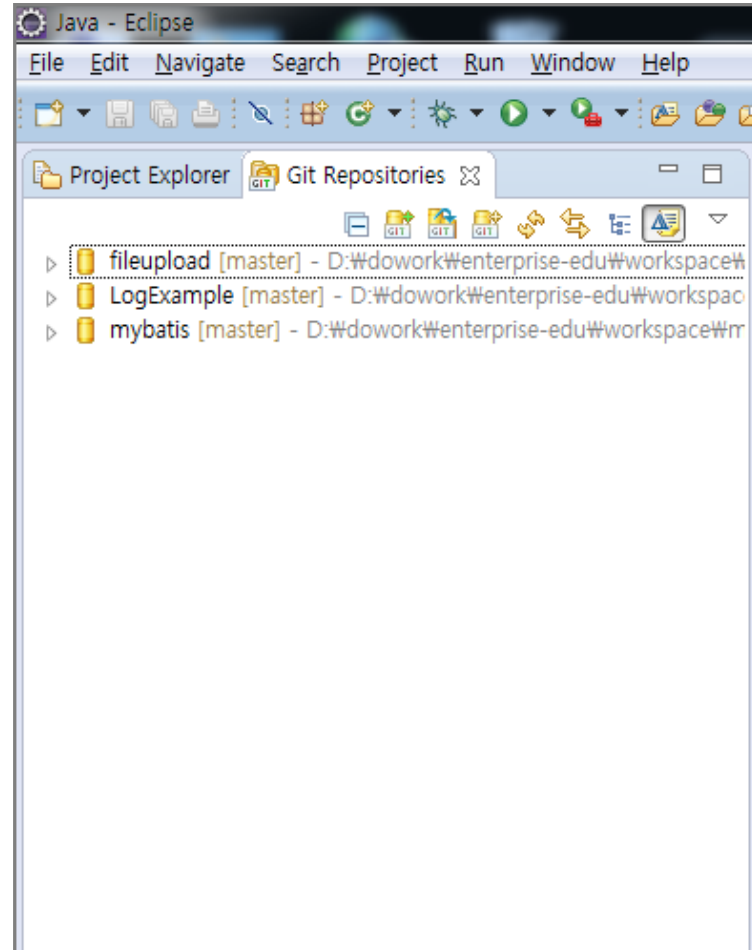
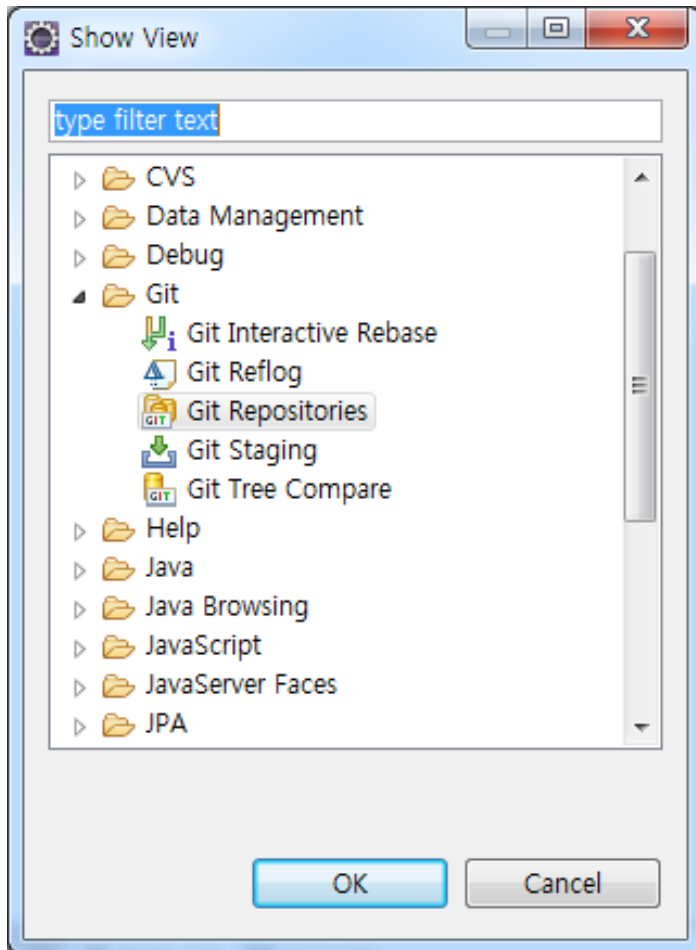
1. Remote와 연결
2. Commit & push
3. Pull
4. Branch checkout

목표

1. Local 컴퓨터에서 작업한 Java project를 github repository에 연결한다.
2. Github repository를 local 컴퓨터로 연결한다 - clone

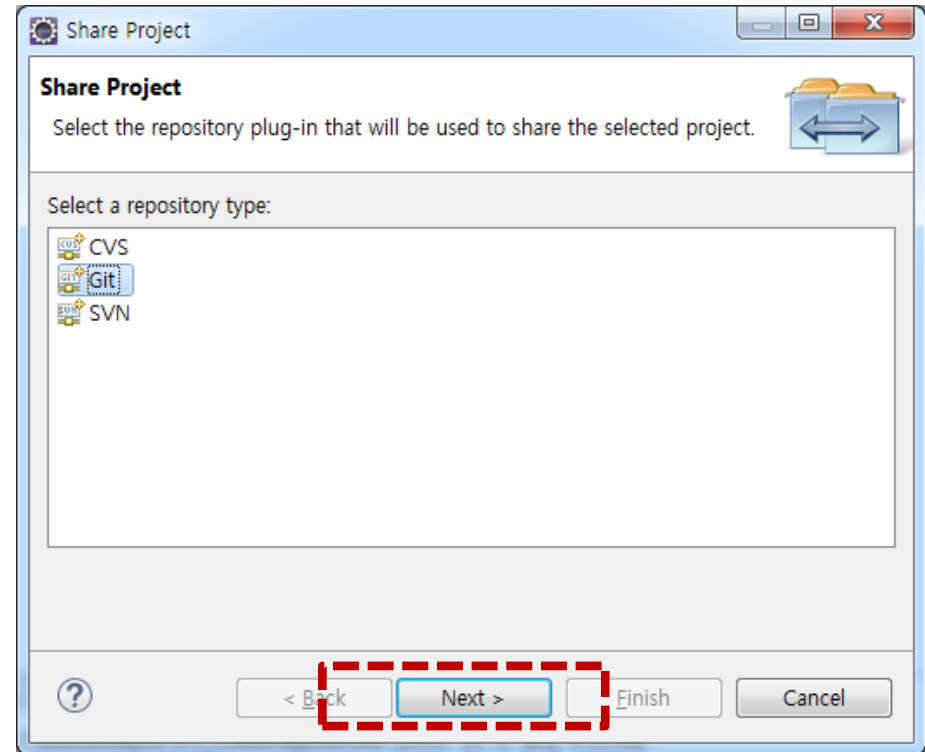
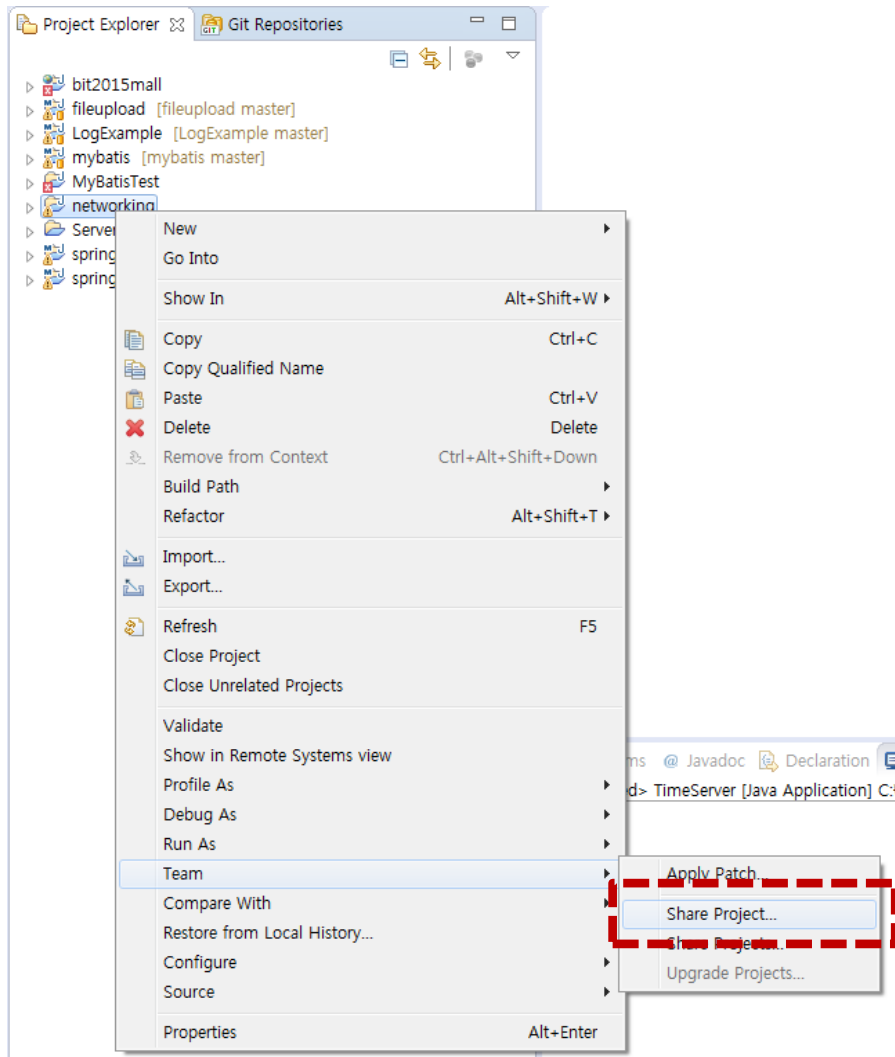
Git Repositories 뷰 꺼내오기

- Window > Show View > Other



기존 프로젝트에서 Team-Share Project

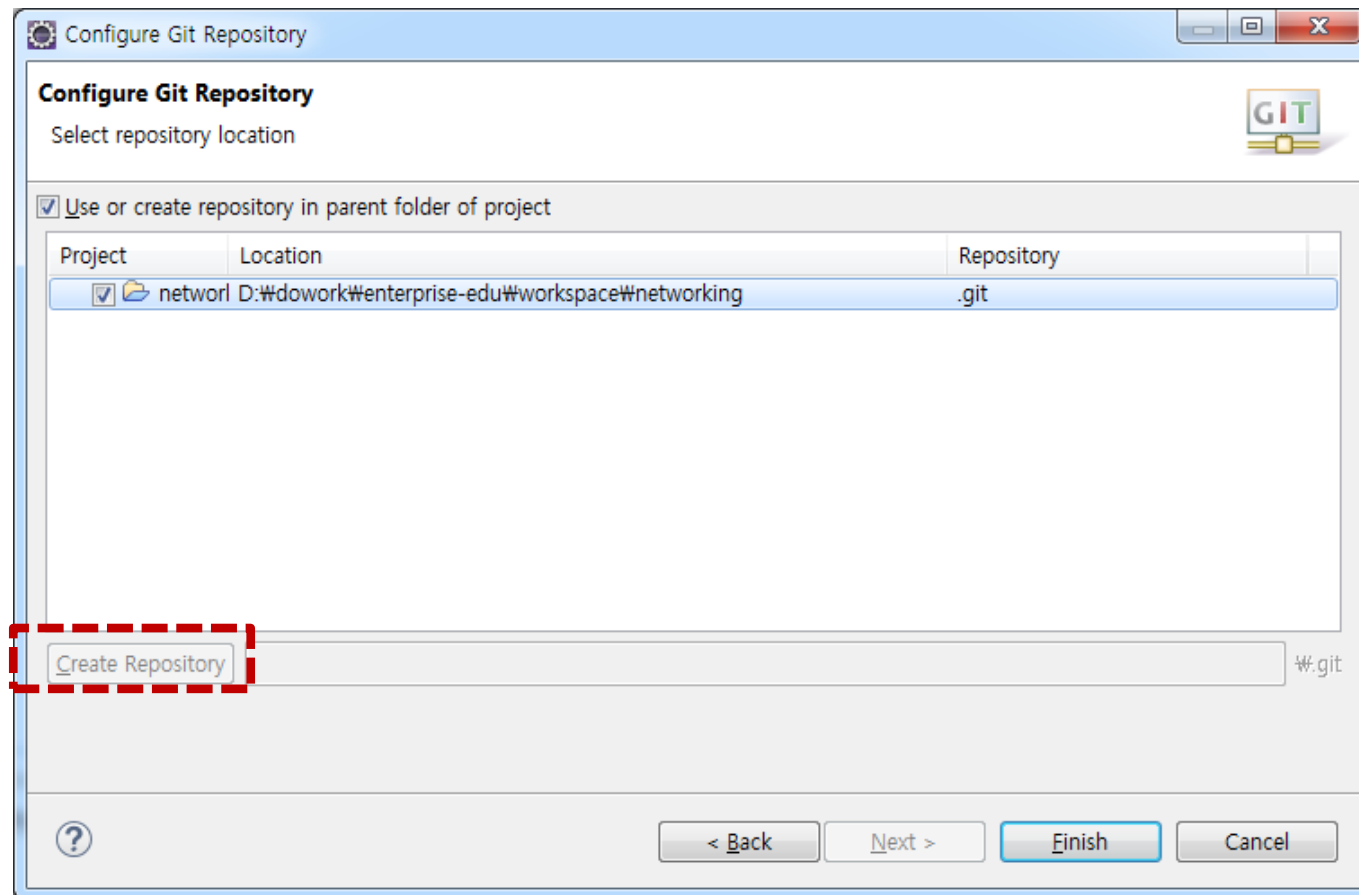
Remote 연결



Local git repository 설정

Remote 연결

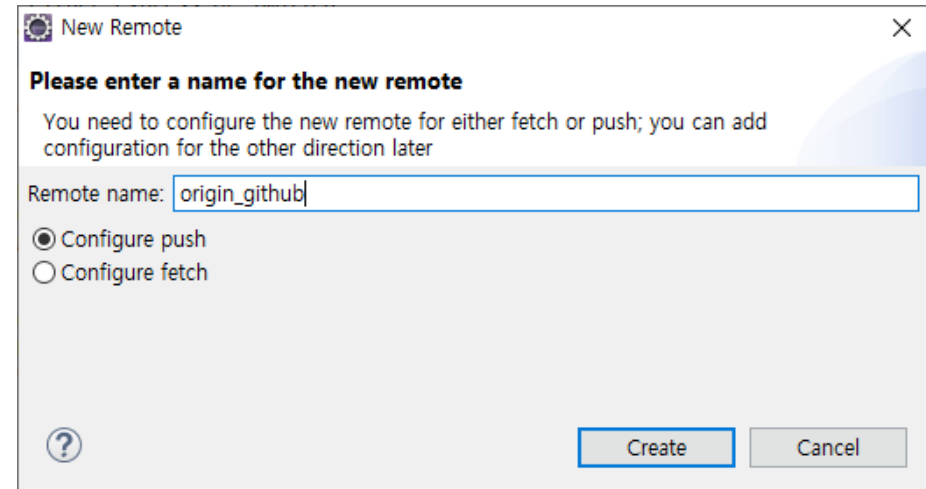
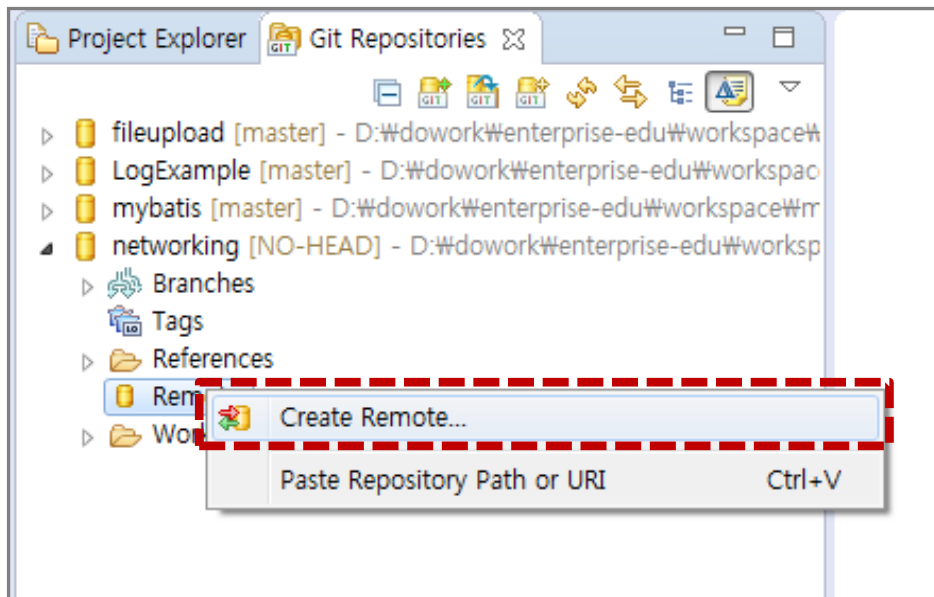
- 따로 로컬 레포지토리를 생성하지 않고 이클립스 프로젝트를 local git repository로 사용



원격 레포지토리 생성 (1/3)

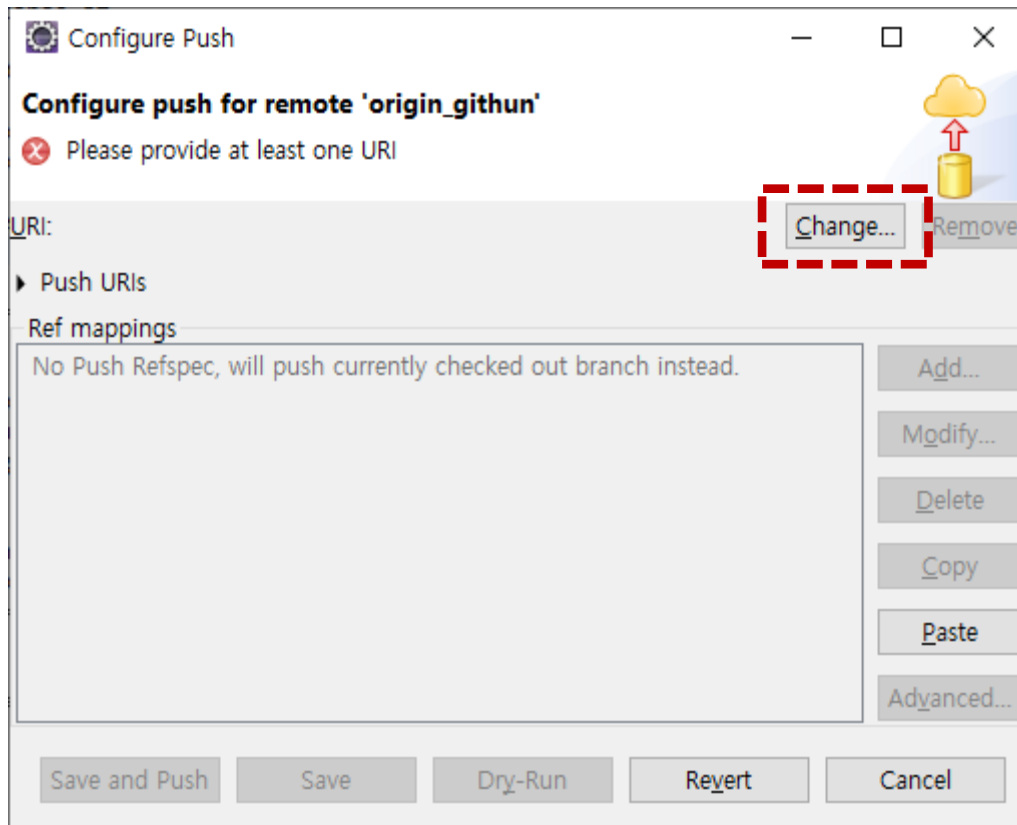
Remote 연결

- 원격 레포지토리를 (github.com) 과 연결한다.
- <http://github.com> 에서 새 레포지토리를 만든다.
 - 비어있는 repository와 연결 - readme.md도 만들지 말것.(나중에 추가하자)
- Remote (원격 레포지토리)의 이름을 정한다. (default- origin)



원격 레포지토리 생성 (2/3)

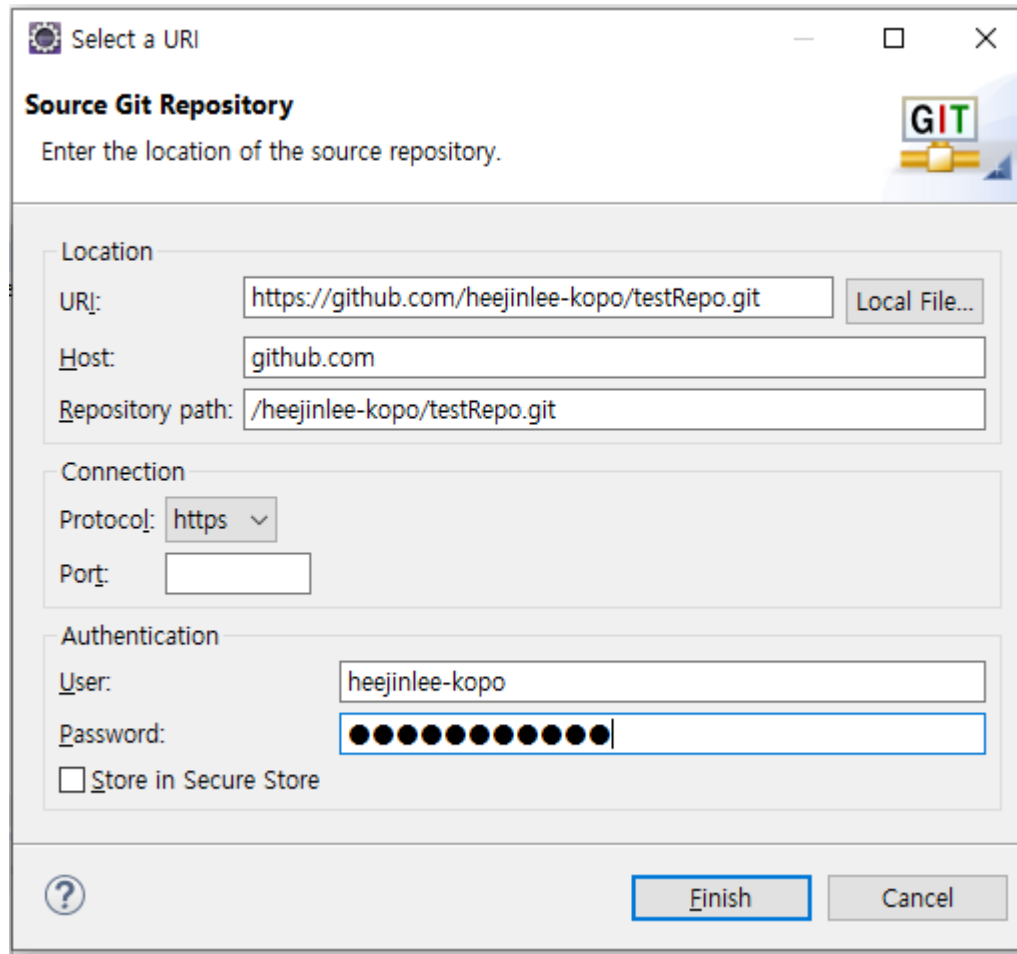
- Origin이 가리키는 원격 정보 주소를 넣는다.
 - Github의 repository – <http://github.com/heejinlee-kopo/testRepo.git>



원격 레포지토리 생성 (3/3)

Remote 연결

- 미리 github에 repository가 존재해야 함



The screenshot shows a Windows-style dialog box titled "Select a URI". Inside, there's a section "Source Git Repository" with a Git logo and the instruction "Enter the location of the source repository." The dialog is divided into three main sections: "Location", "Connection", and "Authentication".

Location:

- URI:
- Host:
- Repository path:

Connection:

- Protocol: (dropdown arrow)
- Port:

Authentication:

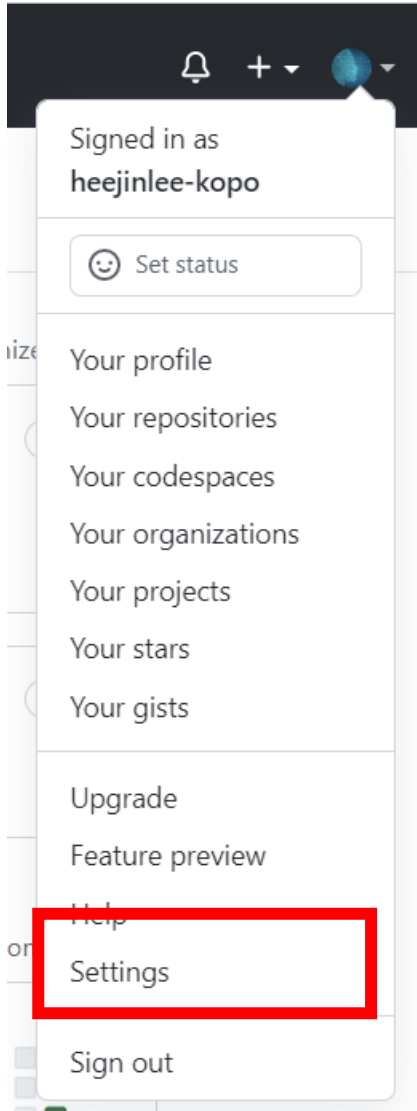
- User:
- Password:
- ☐ Store in Secure Store

At the bottom, there are three buttons: a help button (question mark icon), an "Finish" button, and a "Cancel" button.

Error : xxx.git: not authorized

- Github.com에서 보안정책 변경 - Access token을 설정하자.

<https://joytk.tistory.com/58>



Commit 하지 말아야 할 것들

- Commit 하지 말아야 할 것들

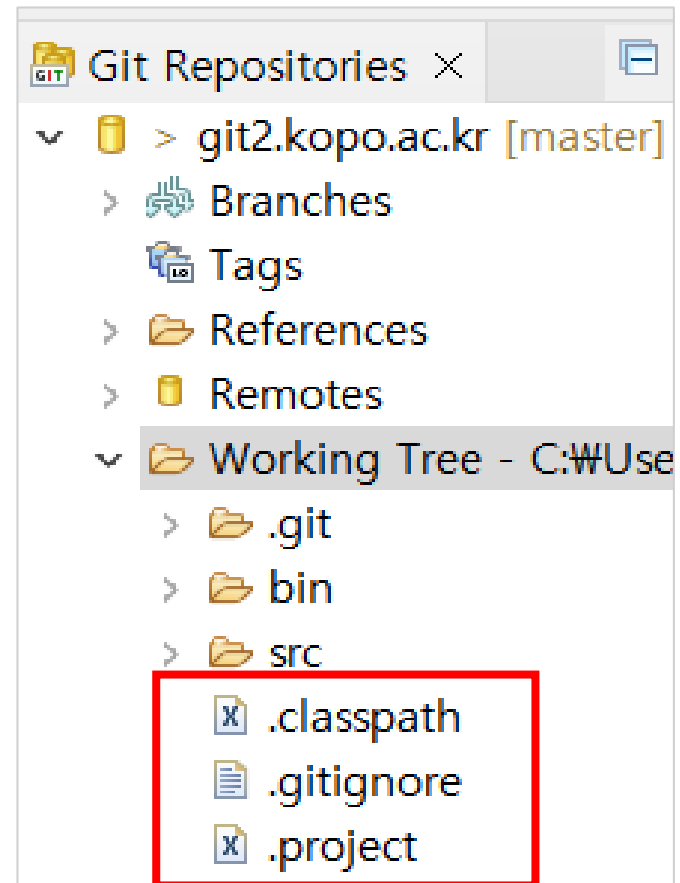
- eclipse 설정
- local 컴퓨터의 설정
- 보안과 관련된 정보
- 불필요한 것들
- 너무 크기가 큰 것들

- Commit 하지 않을 file들은 .gitignore 를 열고 설정

- .gitignore는 나중에도 계속 사용하므로 commit 하자

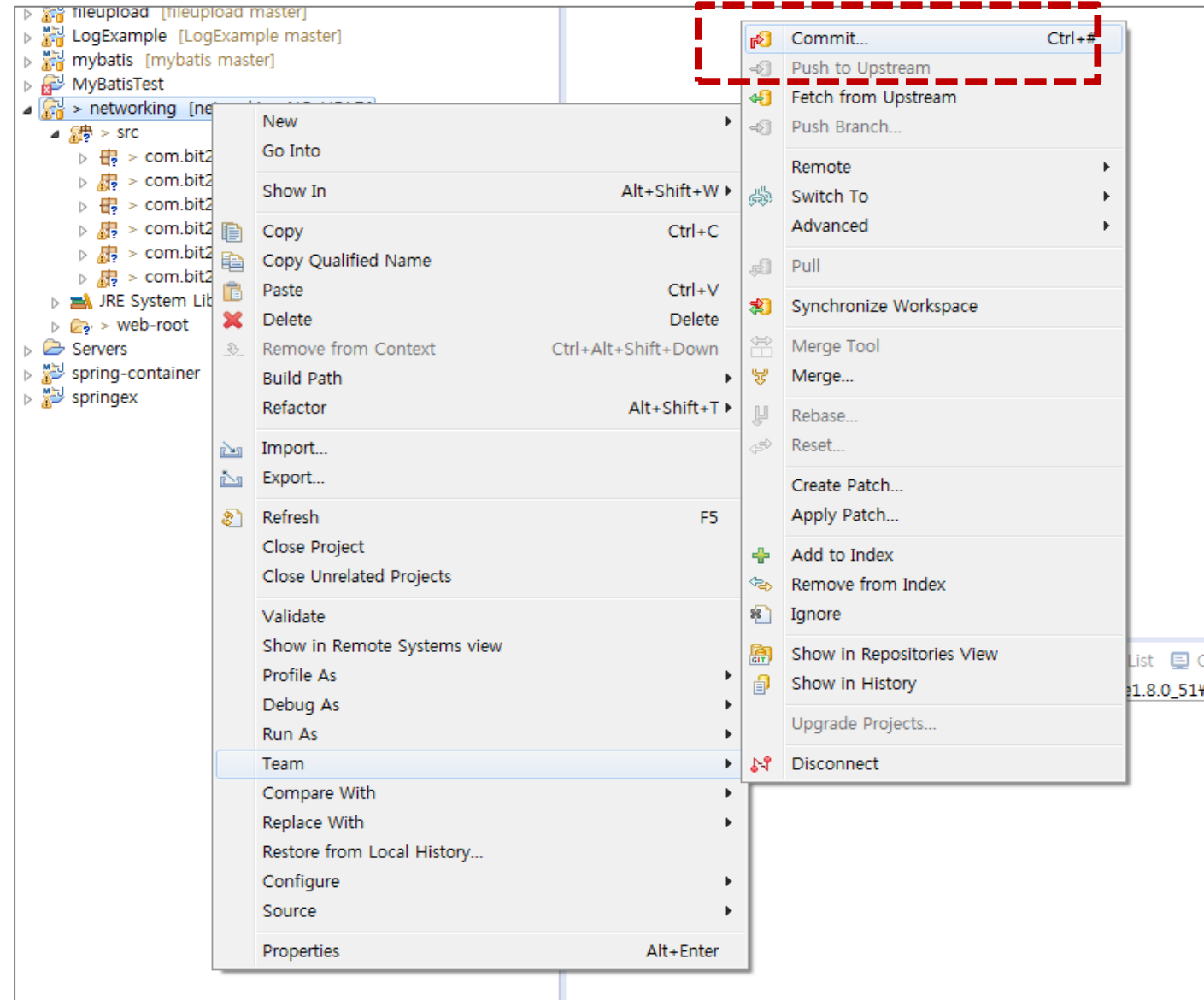
.gitignore 설정

```
/bin/  
/.settings/  
.classpath  
.project
```



Commit and push

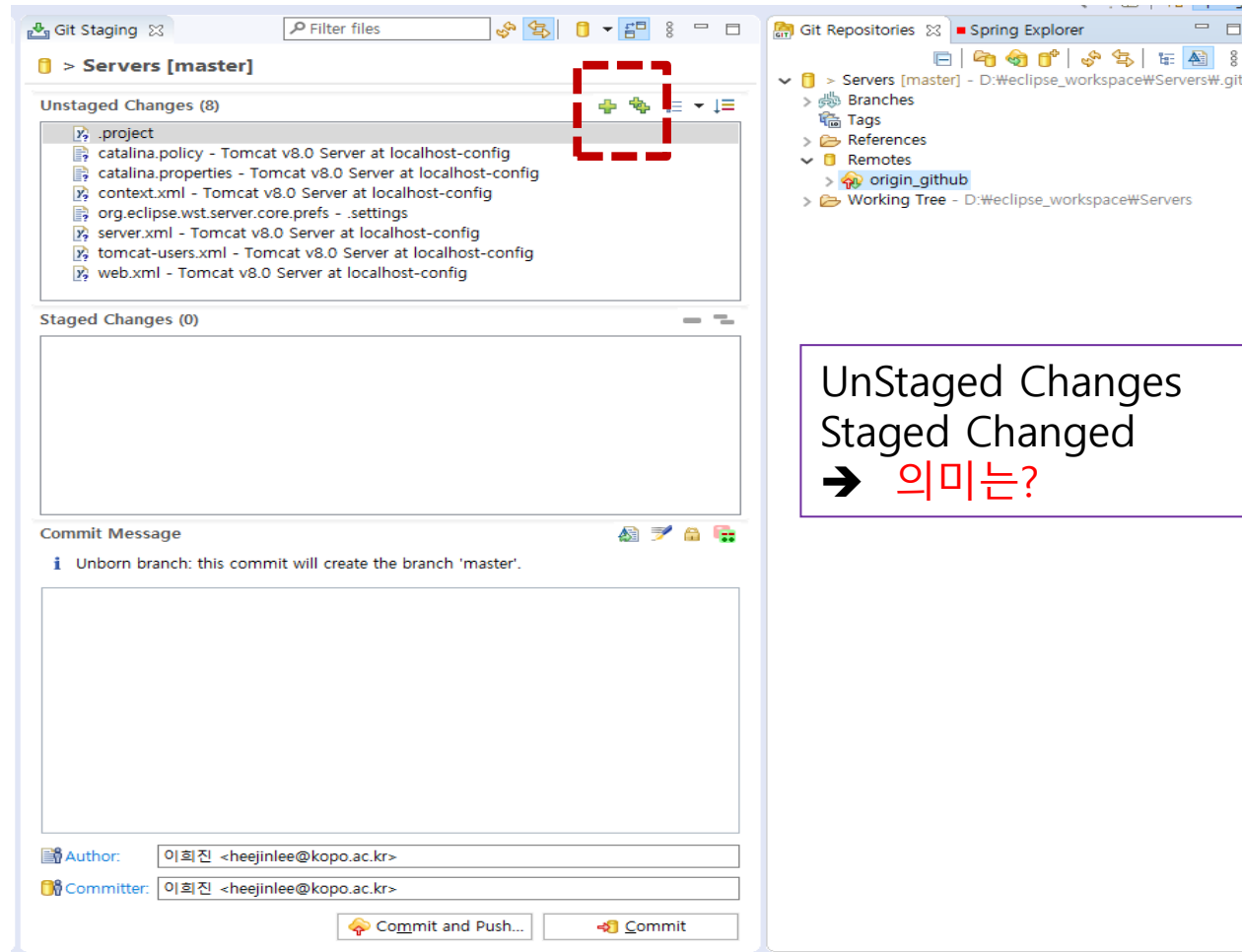
- 기존 code를 commit
- Team-Commit



Unstaged/Staged Changes (1/2)

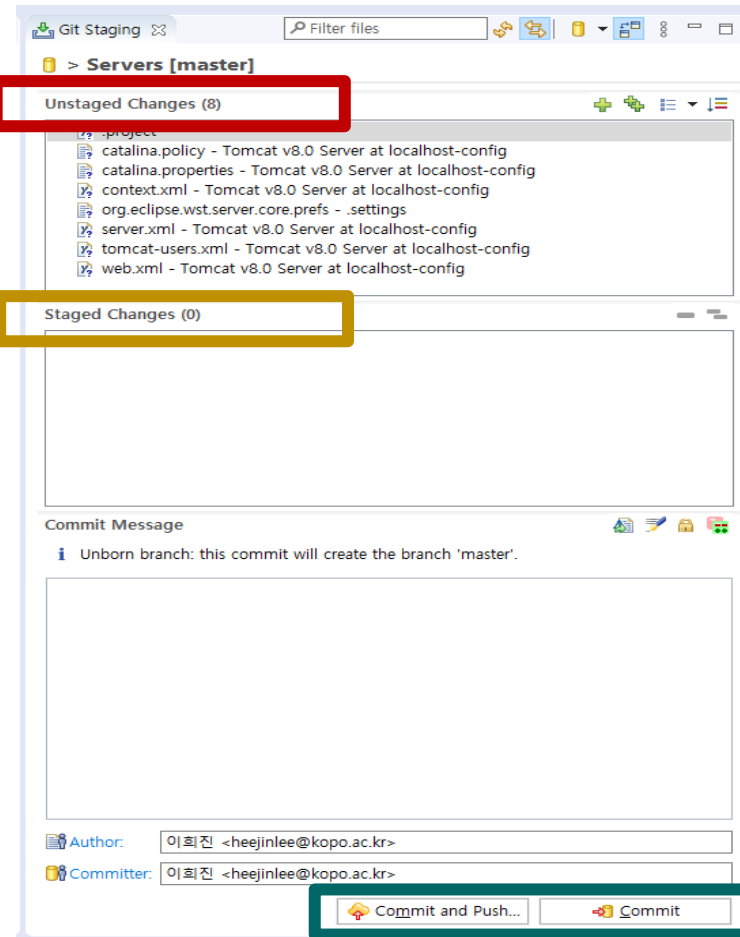
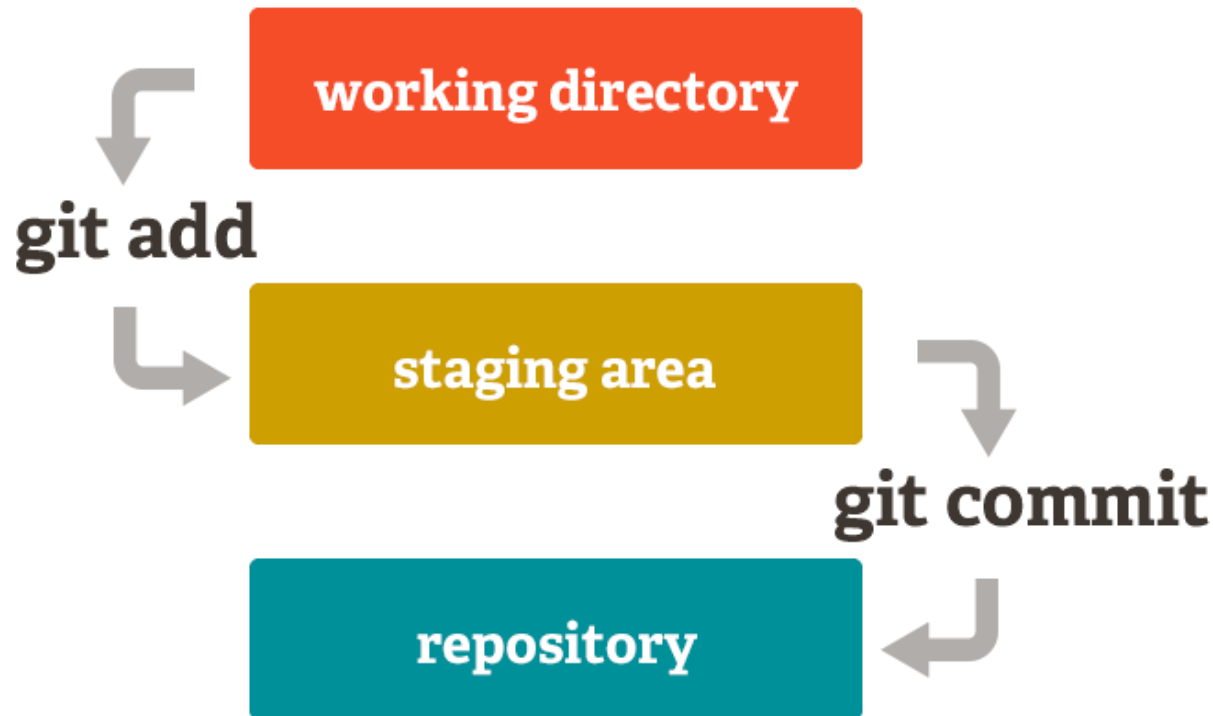
Commit & Push

- Window "Git Staging"
- Window "Git Repositories"



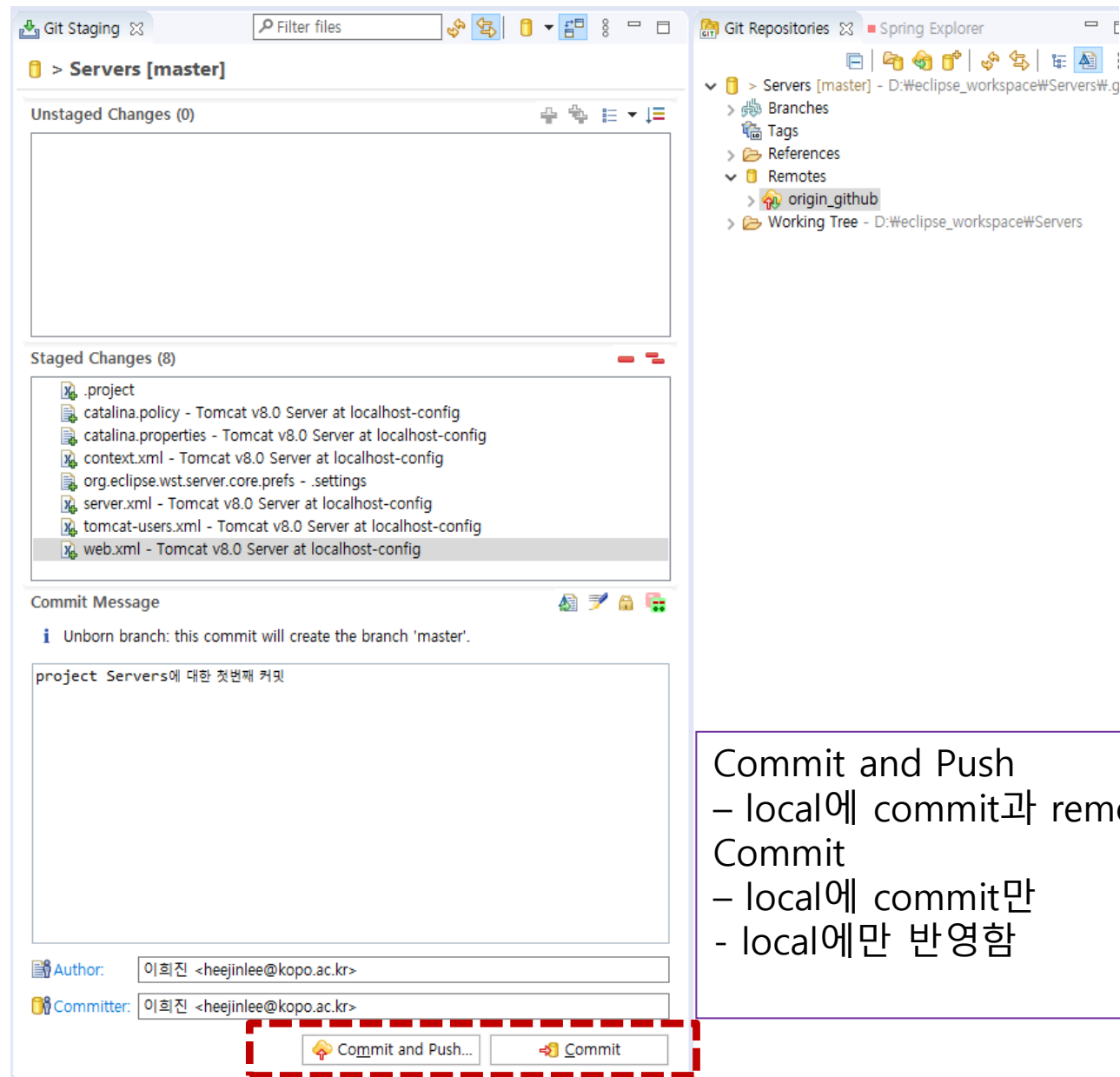
Unstaged/Staged Changes (2/2)

Commit & Push



Commit / Commit and Push

Commit & Push

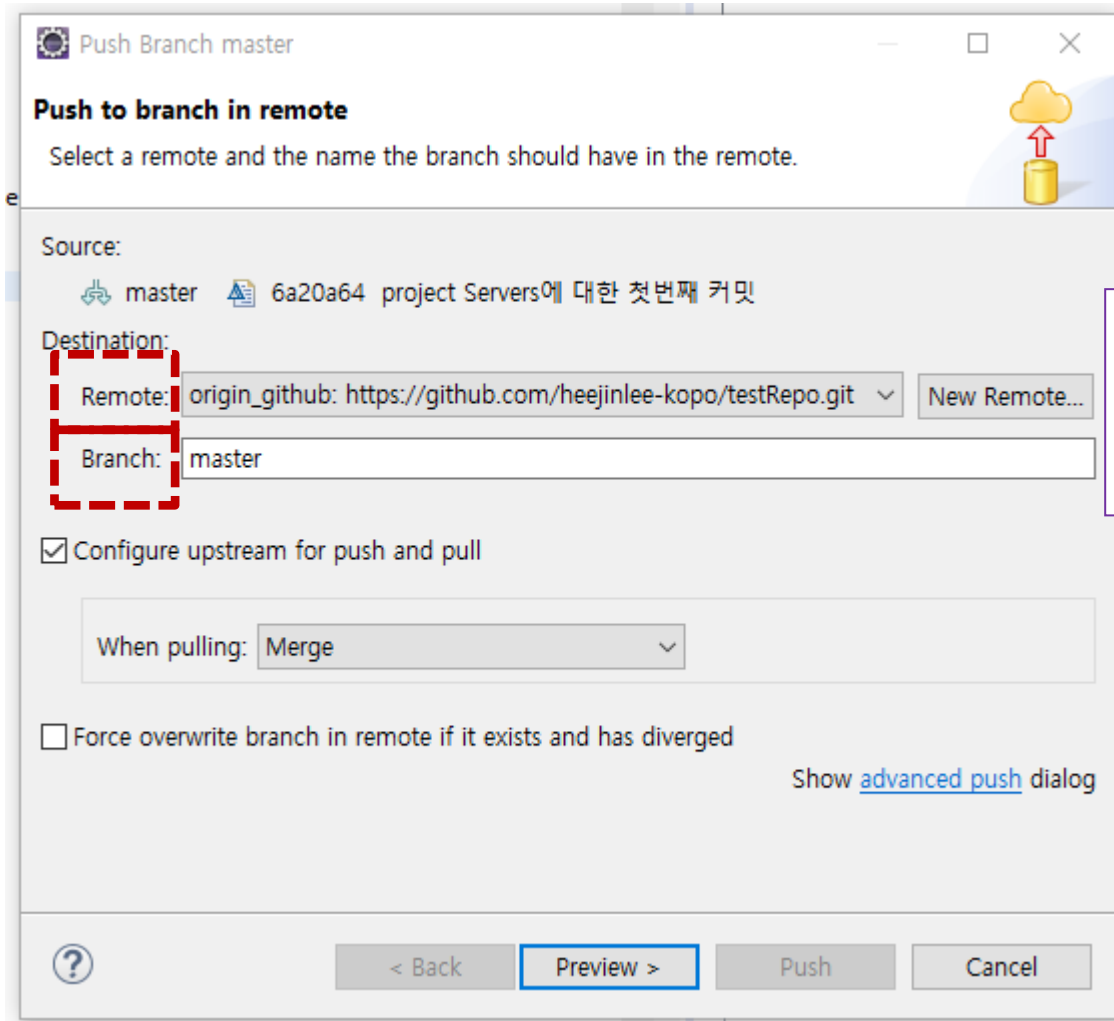


Commit and Push

- local에 commit과 remote에 push를 동시에
- Commit
- local에 commit만
- local에만 반영함

Push – remote 에 반영

Commit & Push



Remote에 반영하자!!

⇒ Remote에 master branch를 새로 만들어서 push하는 것

Remote

- 반영할 외부의 github repository

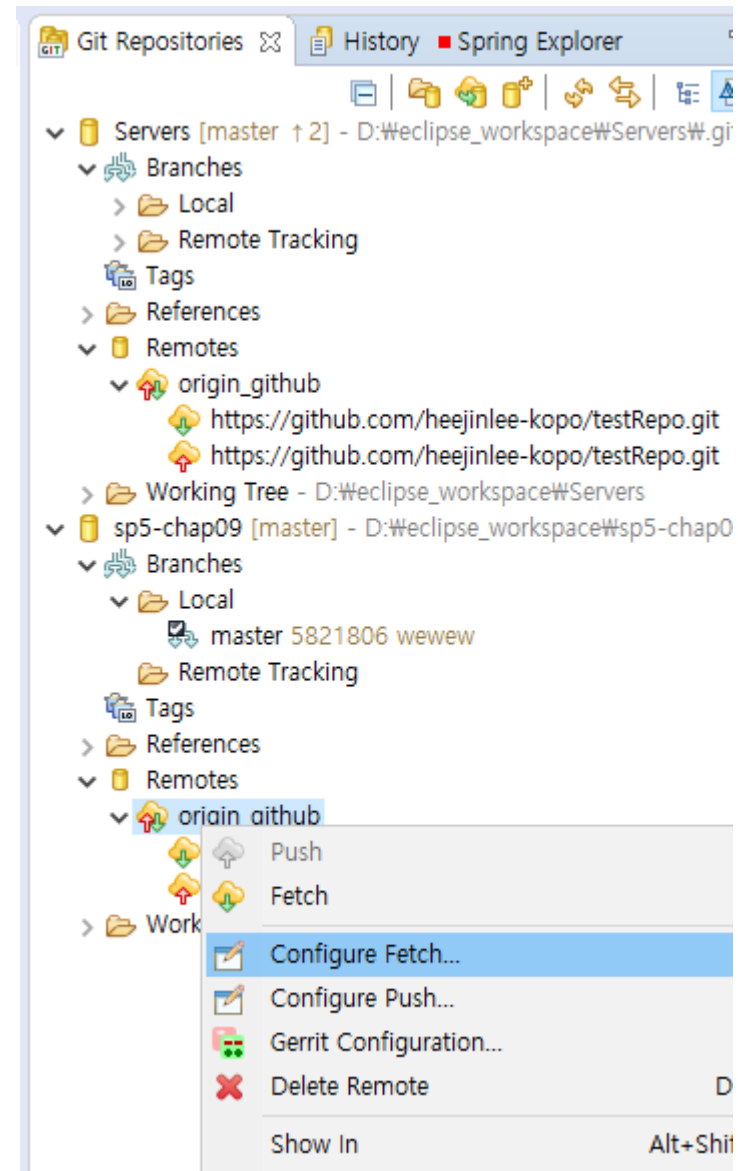
Branch

- 반영할 외부 github의 repository의 branch

Fetch - configure fetch (1/5)

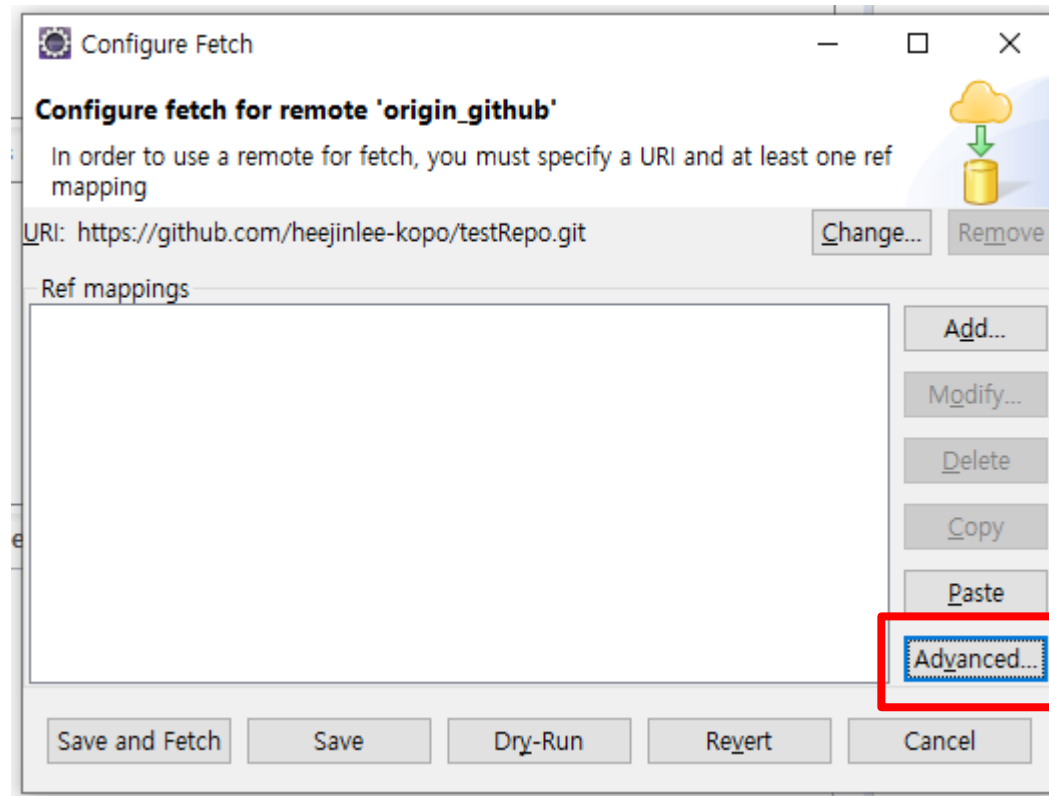
Push confirmation error

- Window "Git Repositories"
 - Remotes->origin_github-> Configure Fetch



Fetch - configure fetch (2/5)

Push confirmation error



Fetch - configure fetch (3/5)

Push confirmation error

Source ref:
refs/heads/master

Destination ref:
refs/remotes/"외부_repository_이름"/master

Source ref:
refs/heads/*

Destination ref:
refs/remotes/"외부_repository_이름"/*

Configure Fetch

Fetch Ref Specifications
Select refs to fetch.

Add create/update specification

Source ref: Destination ref:

Add predefined specification

Destination Ref	Force Update	Remove

Annotated tags fetching strategy

☒ Automatically follow tags if we fetch the thing they point at

☐ Always fetch tags, even if we do not have the thing it points at

☐ Never fetch tags, even if we have the thing it points at

☐ Save specifications in 'origin_github' configuration

Fetch - configure fetch (4/5)

Push confirmation error

Configure Fetch

Fetch Ref Specifications
Select refs to fetch.

Add create/update specification

Source ref: Destination ref:

Add predefined specification

Specifications for fetch

Source Ref	Destination Ref	Force Update	Remove
refs/heads/master	refs/remotes/origin_github/...	<input type="checkbox"/>	

Annotated tags fetching strategy

☒ Automatically follow tags if we fetch the thing they point at

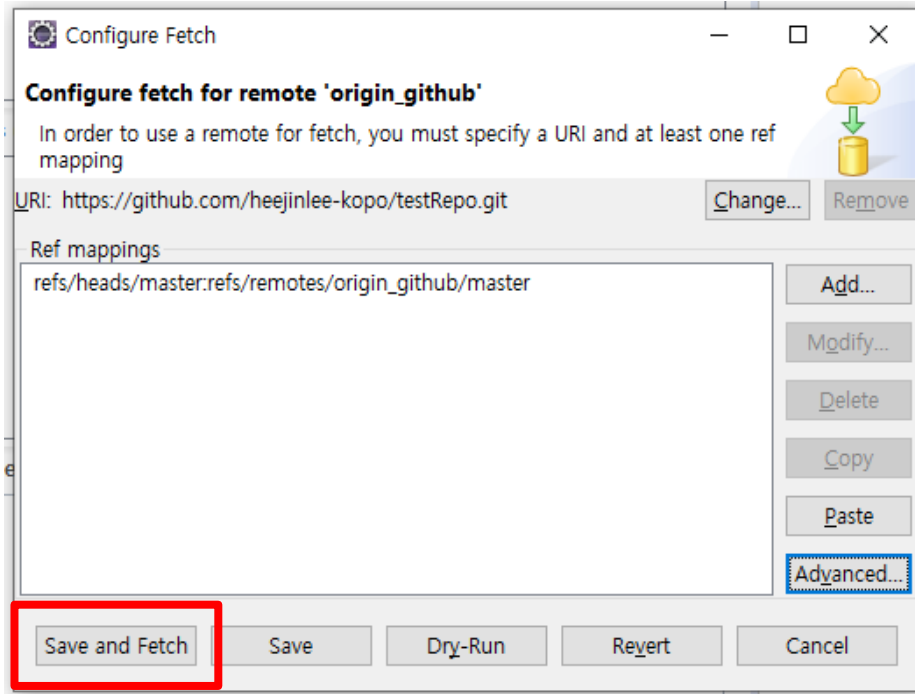
☐ Always fetch tags, even if we do not have the thing it points at

☐ Never fetch tags, even if we have the thing it points at

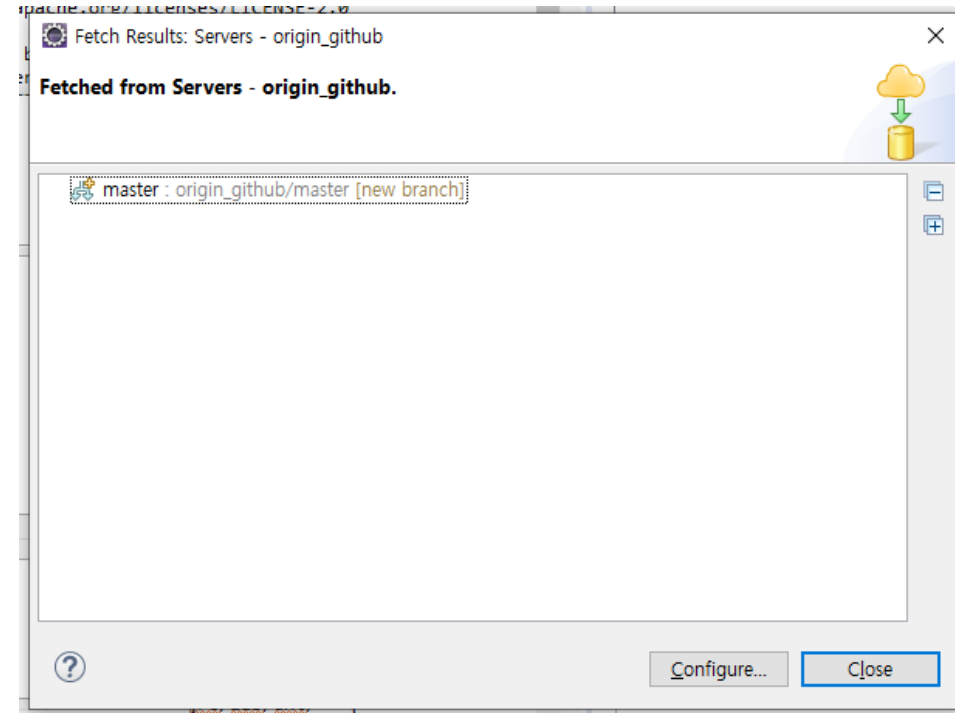
☐ Save specifications in 'origin_github' configuration

Fetch - configure fetch (5/5)

Push confirmation error

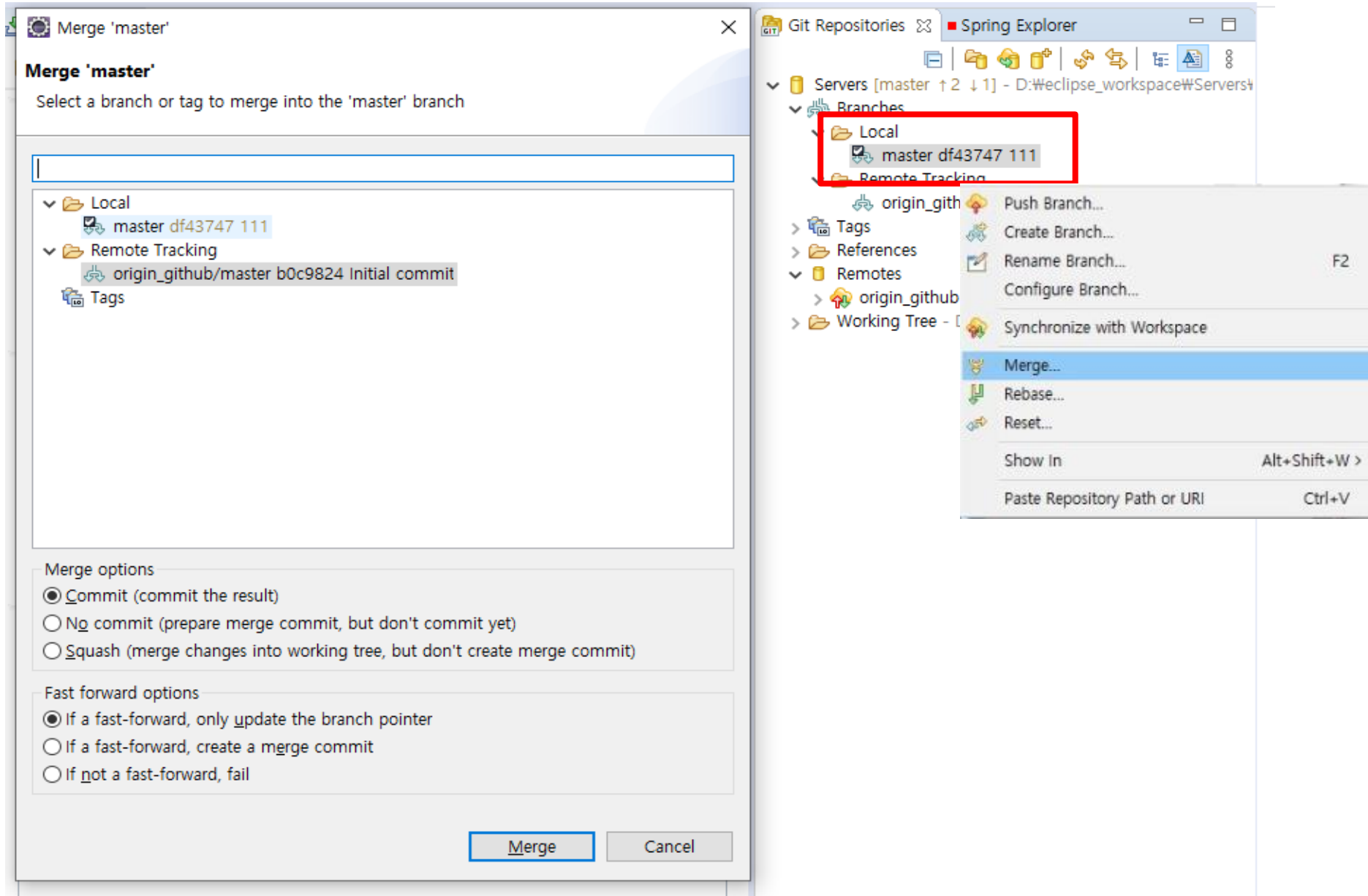


Save and Fetch



Fetch한 후에 merge해야 pull 완성

Merge

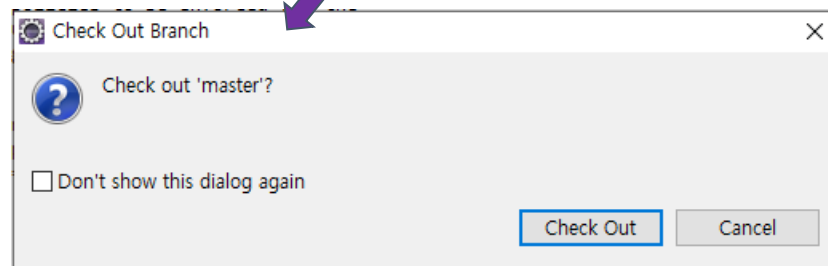
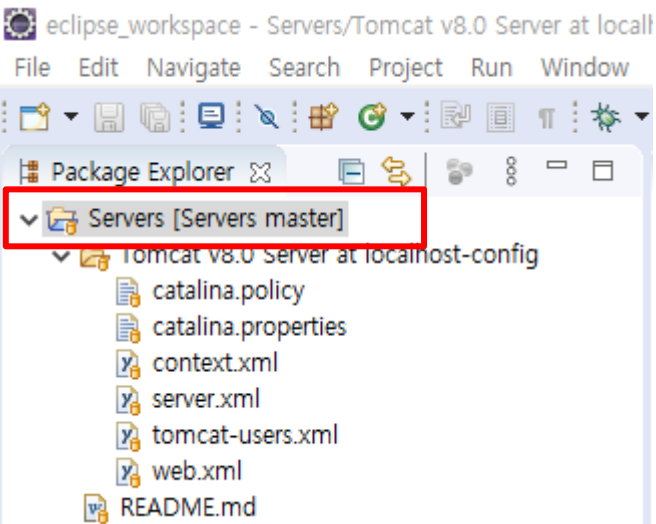
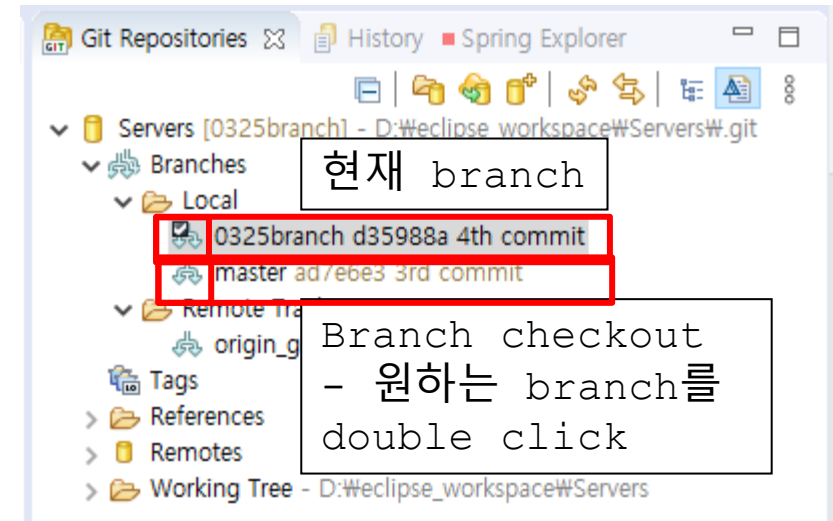
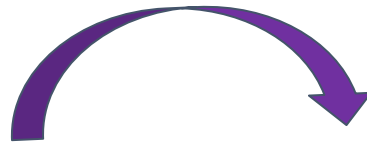
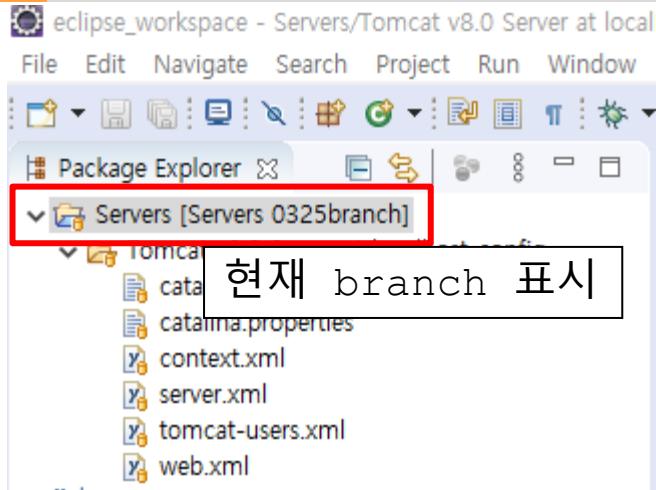


The background of the slide features abstract geometric shapes in shades of orange and blue. A dark blue horizontal band is positioned across the upper middle section, serving as a backdrop for the title text.

Branch 변경

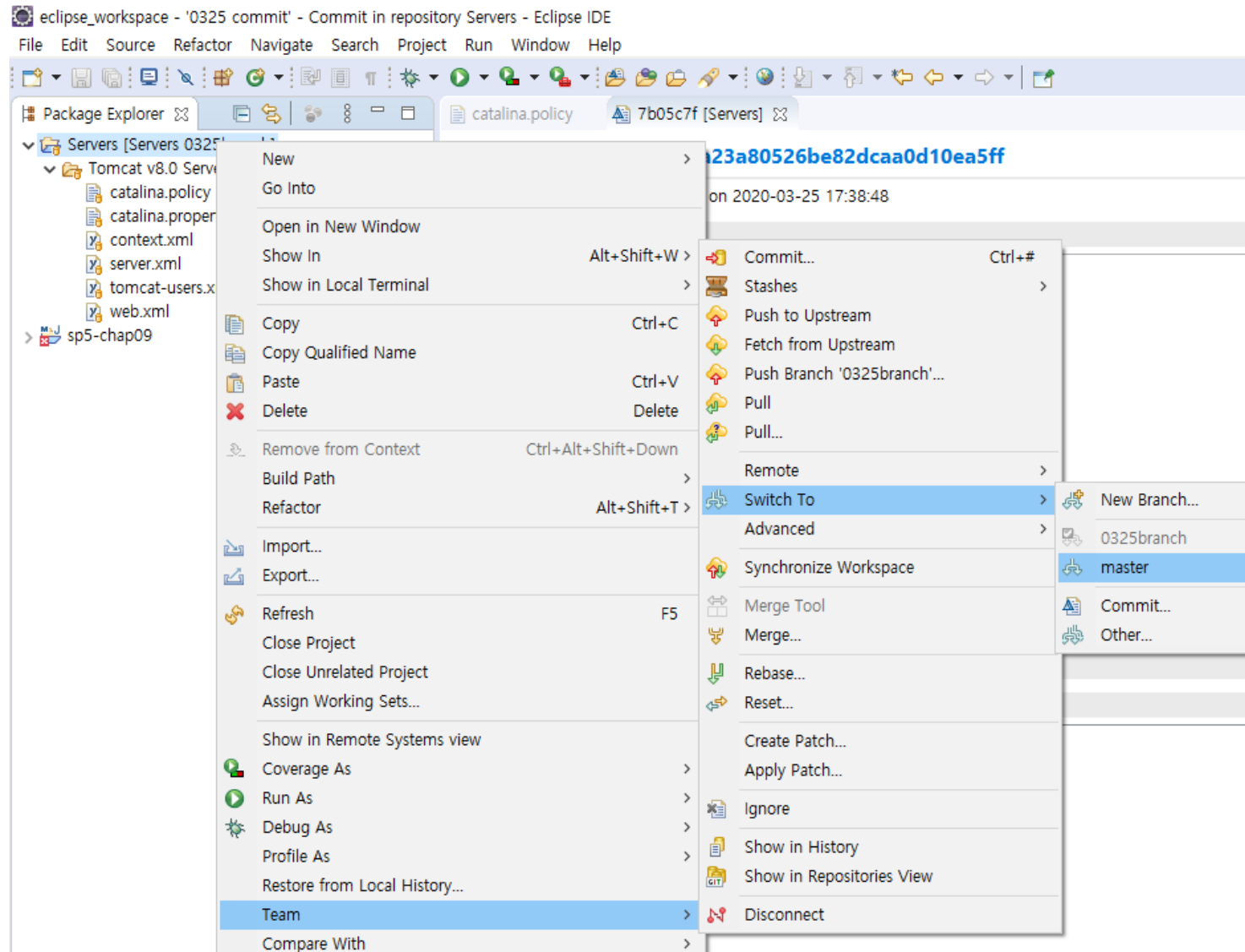
Branch check out – 방법1

Branch checkout



Branch check out – 방법2

Branch checkout



Q1. 윈도우 탐색기에서는 branch 어떻게 표현??

Branch checkout

- 특정 위치에 여러 개의 branch가 같이 존재한다.
 - C:\workspace\HelloWorld
 - 총 branch 3개가 존재 – branch1, branch2, branch3
- Eclipse에서 branch checkout하면 각 branch마다 파일이 다른 것이 보이는데, 윈도우 탐색기에서는 branch 마다 다른걸 어떻게 나타내지?

A1. 윈도우 탐색기는 현재 branch를 보여줌

Branch checkout

- Eclipse에서 현재 checkout 되어 있는 branch를 보여준다.

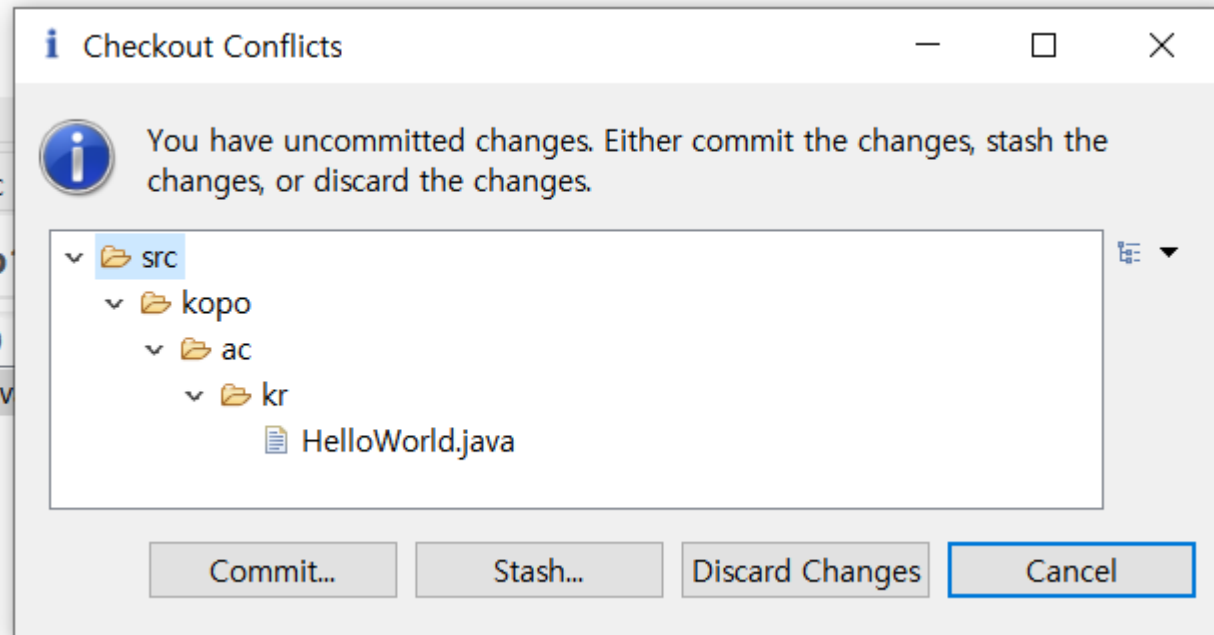
➔ 확인해보자

- Eclipse와 윈도우 탐색기를 동시에 띄우고, eclipse에서 branch 변경시 윈도우 탐색기의 내용이 branch마다 변경되는 것 확인
- Branch1에서 branch1test.txt를 생성 후 branch1에만 commit, Branch2에서 branch1test.txt가 없는 것을 확인.
윈도우 탐색기에서도 확인.

Commit 을 하지 않고 branch를 변경하면?

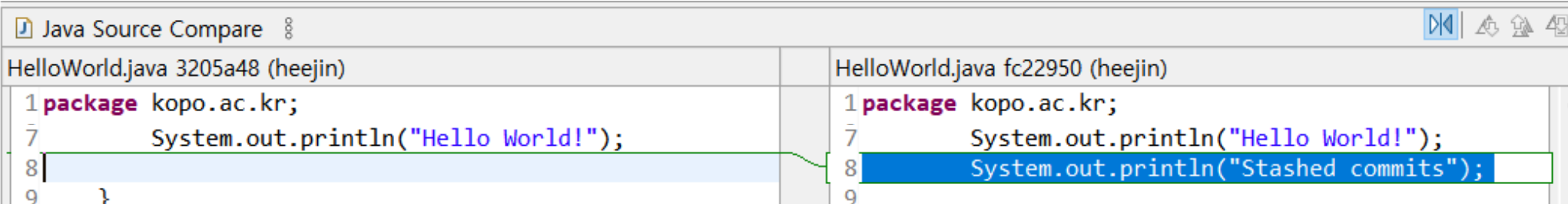
- Q. 소스코드 저장을 하고, commit을 안하고 branch를 바꾸면?
- A. 셋중에 하나 선택.

Commit, Stash (commit하지 않고 임시저장), Discard(버리기)



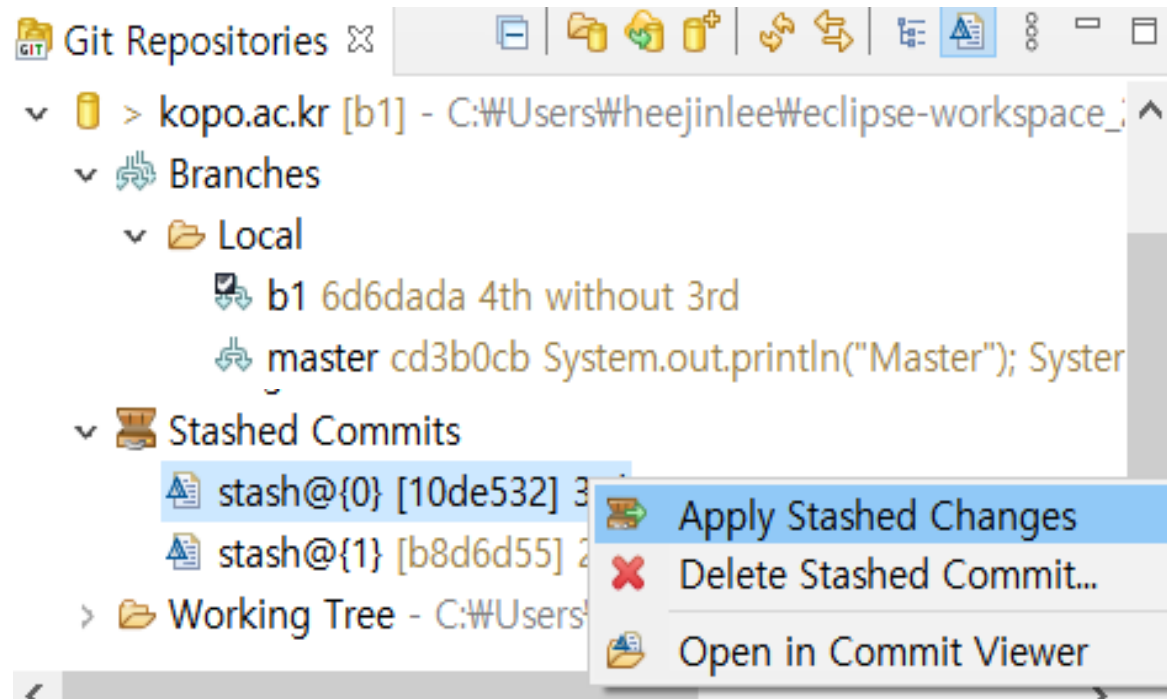
Commit 을 하지 않고 branch를 변경하면? - Stash

- Stash (commit하지 않은 임시 저장) – 나중에 저장된 것을 반영 할 수 있음



```
Java Source Compare ⓘ
HelloWorld.java 3205a48 (heejin) | HelloWorld.java fc22950 (heejin)
1 package kopo.ac.kr; | 1 package kopo.ac.kr;
7     System.out.println("Hello World!"); | 7     System.out.println("Hello World!");
8 | 8     System.out.println("Stashed commits");
9 } | 9
```

- Apply Stashed Changes 선택
- 충돌(Conflict)이 있을 경우에는 직접 사람이 merge 해야 함





Repository clone

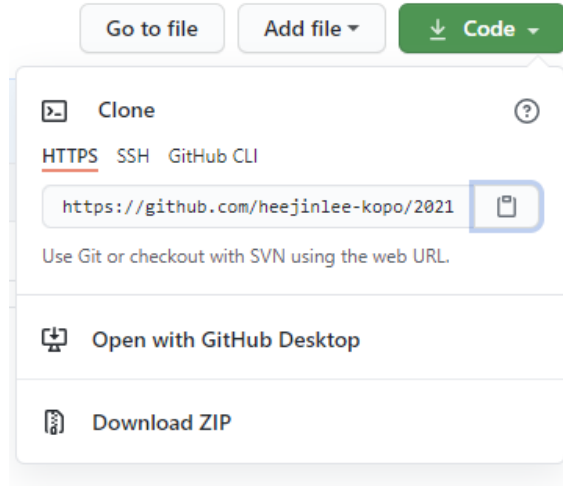
github에서 내 local computer로

Repository Clone

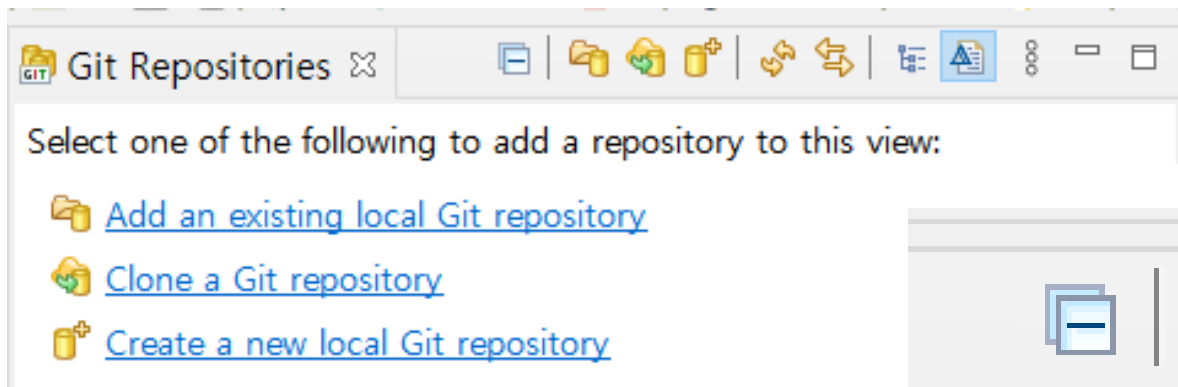
- Github에 존재하는 repository를 내 local computer에 그대로 가져오는 기능
- **활용 상황**
 - 광명융합기술교육원에서 야간 자율학습까지 열심히 과제를 80% 정도 했는데, 집에 가야 하는 경우
 - 모처럼 황금 연휴가 3일이나 생겼는데, 교수님께서 수업시간에 하던 코드를 변형해서 과제를 내라고 하는 경우, 학교는 문을 닫음
 - 노트북이 없어서, 학교의 데스크탑과 내방의 데스크탑만 쓸 수 있는 경우
 - 세계적인 오픈 소스 프로젝트의 코드들은 얼마나 잘 짰는지, 과연 나보다 얼마나 잘하는지 버그들을 찾아 보고 싶은 경우
 - 회사에 처음 들어가서, 선배들이 피땀 흘려가며 만든 코드를 분석하고, 나도 commit을 해야 하는 경우
- **그대로 repository를 가져오면 됨.**
 - Zip으로 가져오는 경우 – 해당 그 snapshot만 가져옴
 - Clone하는 경우 – 모든 branch, commit정보를 가지고 온다.

Git Repositories clone

- 1) github 홈페이지에서 repository 주소를 copy한다.



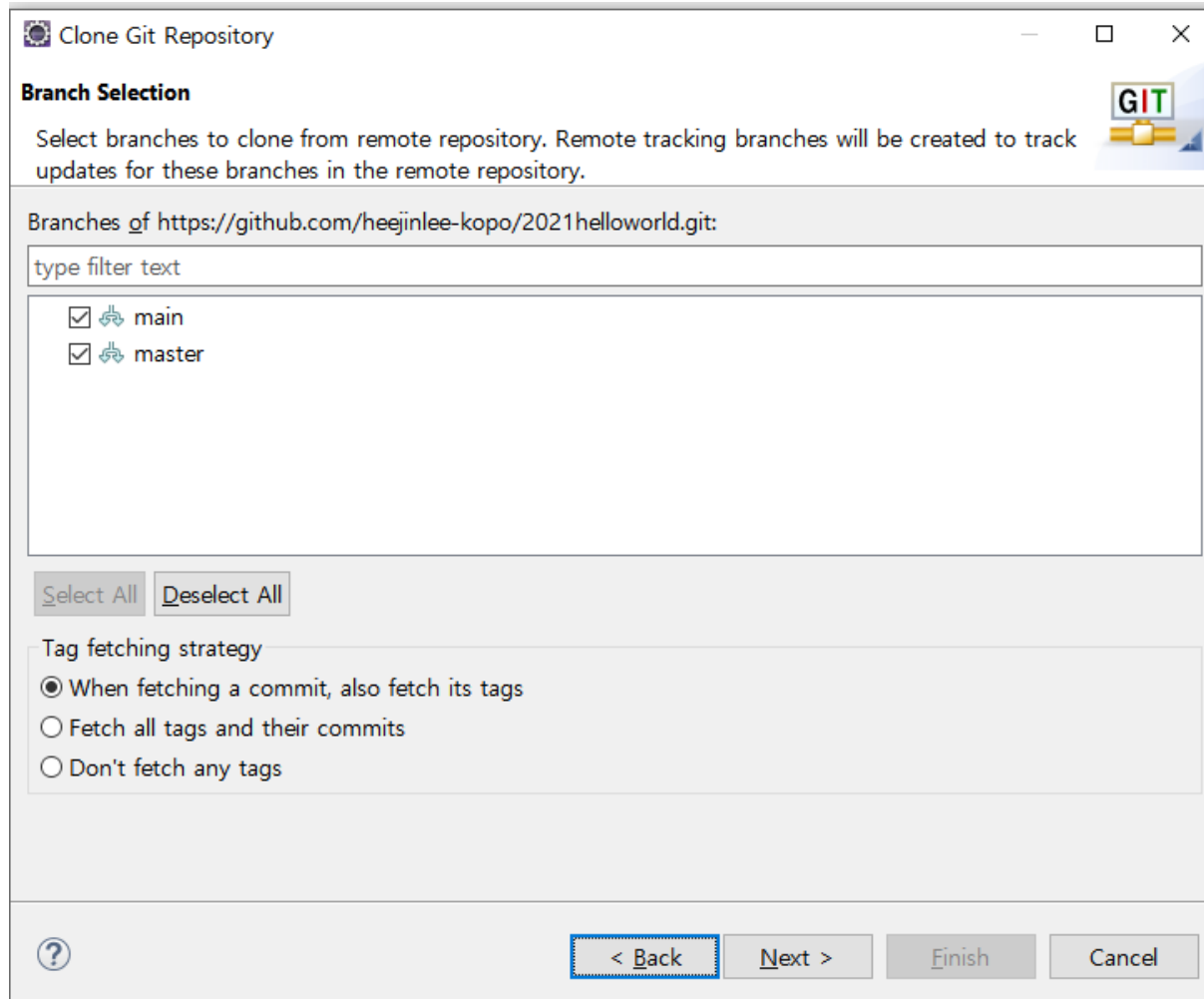
- 2) Git Repositories에서 Clone a Git repository 클릭
=> repository 주소가 자동입력



Git Repositories clone

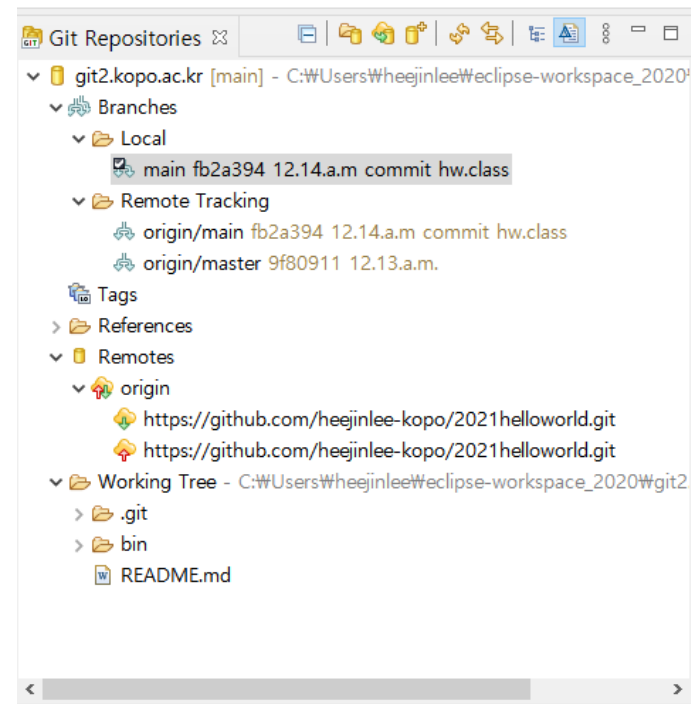
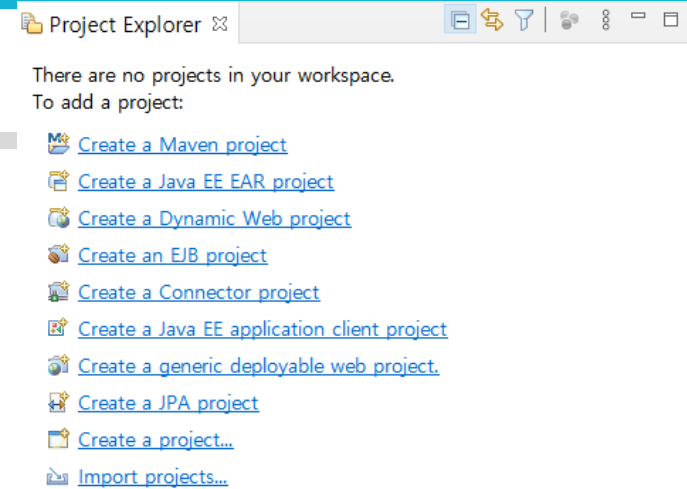
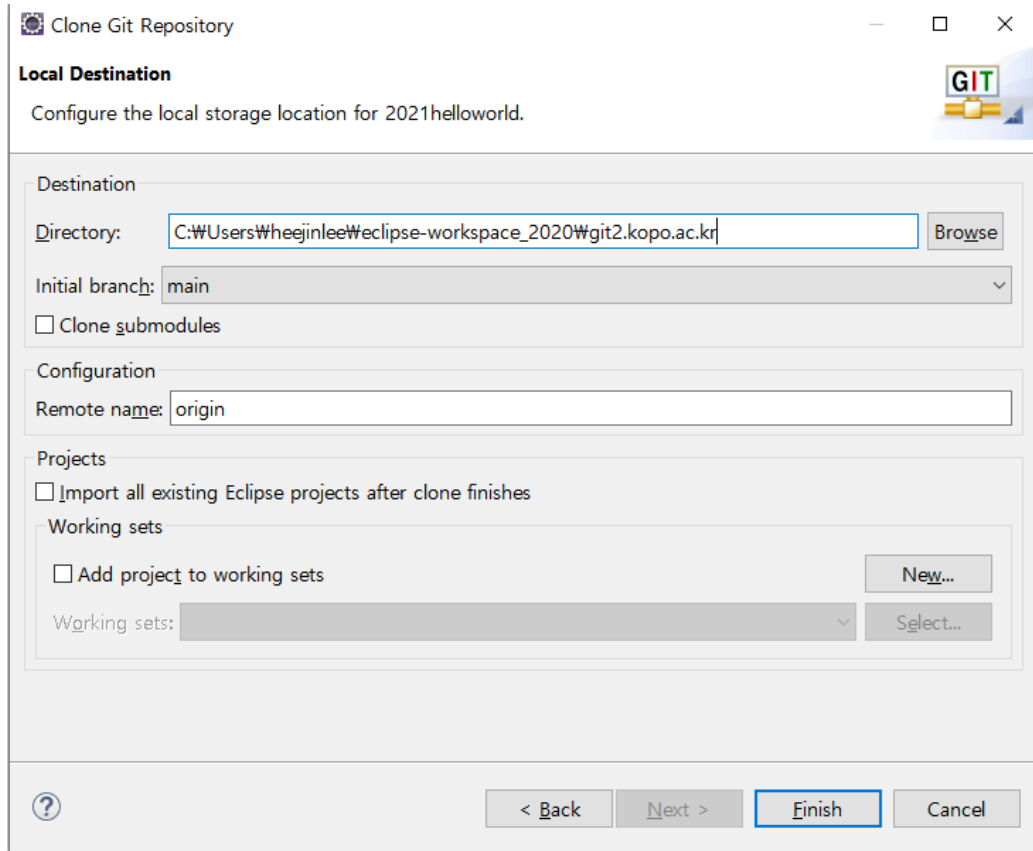
- **Branch Selection**

- 나중에 branch 추가 가능



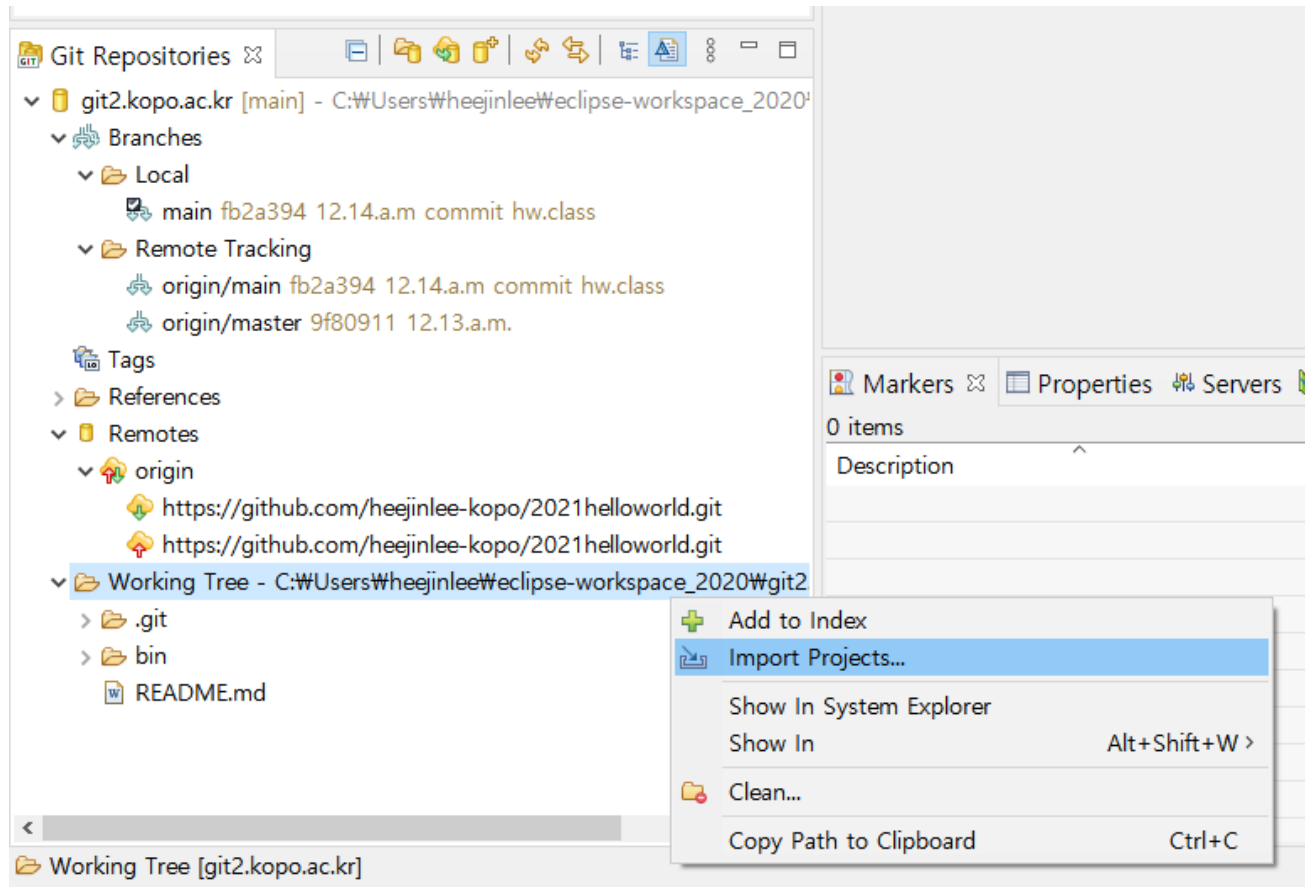
Git Repositories clone

- Destination – clone할 위치 선정
 - Initial branch 선정



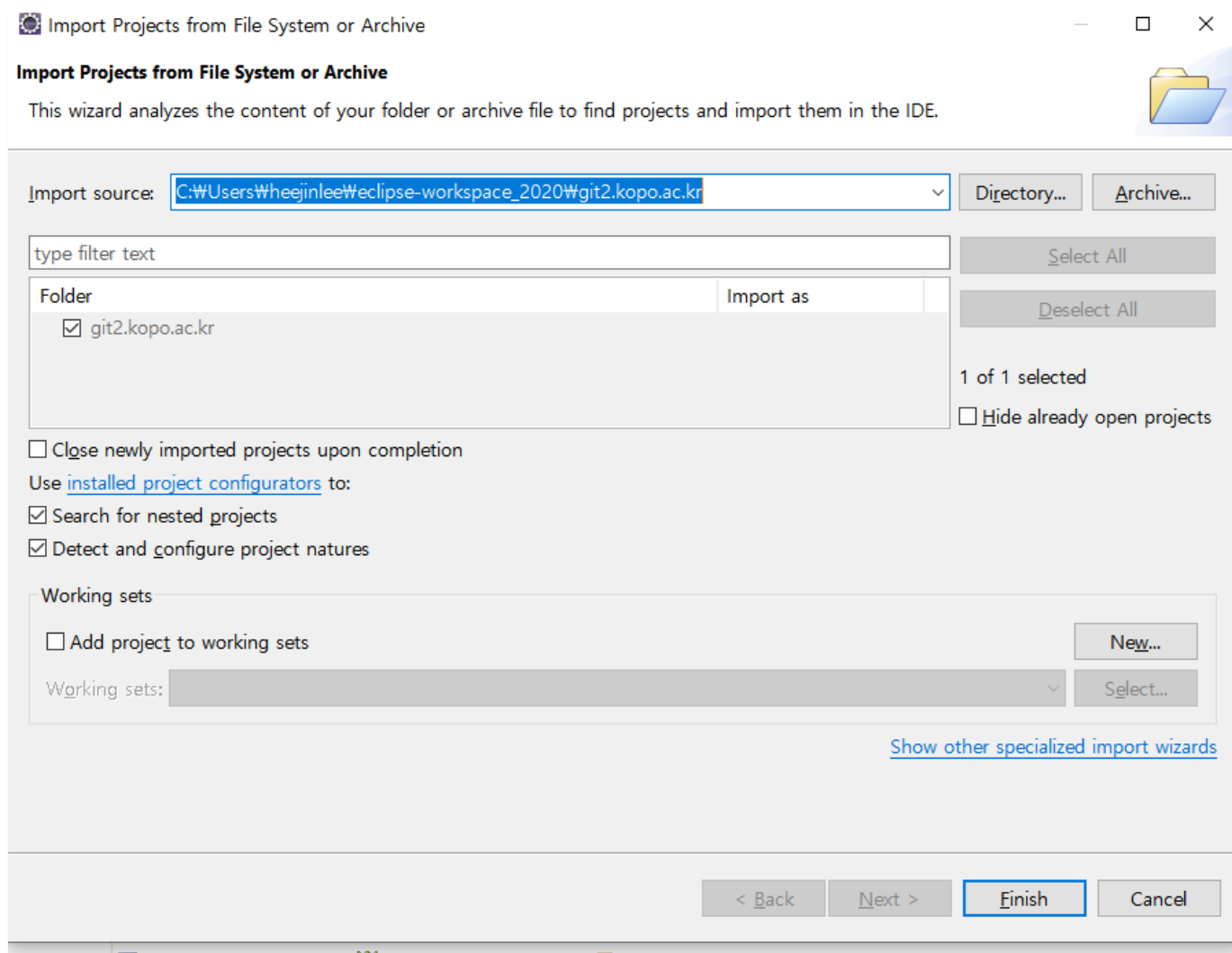
Package Explorer에 package 추가

- Working Tree에서 Import Projects 선택



Package Explorer에 package 추가

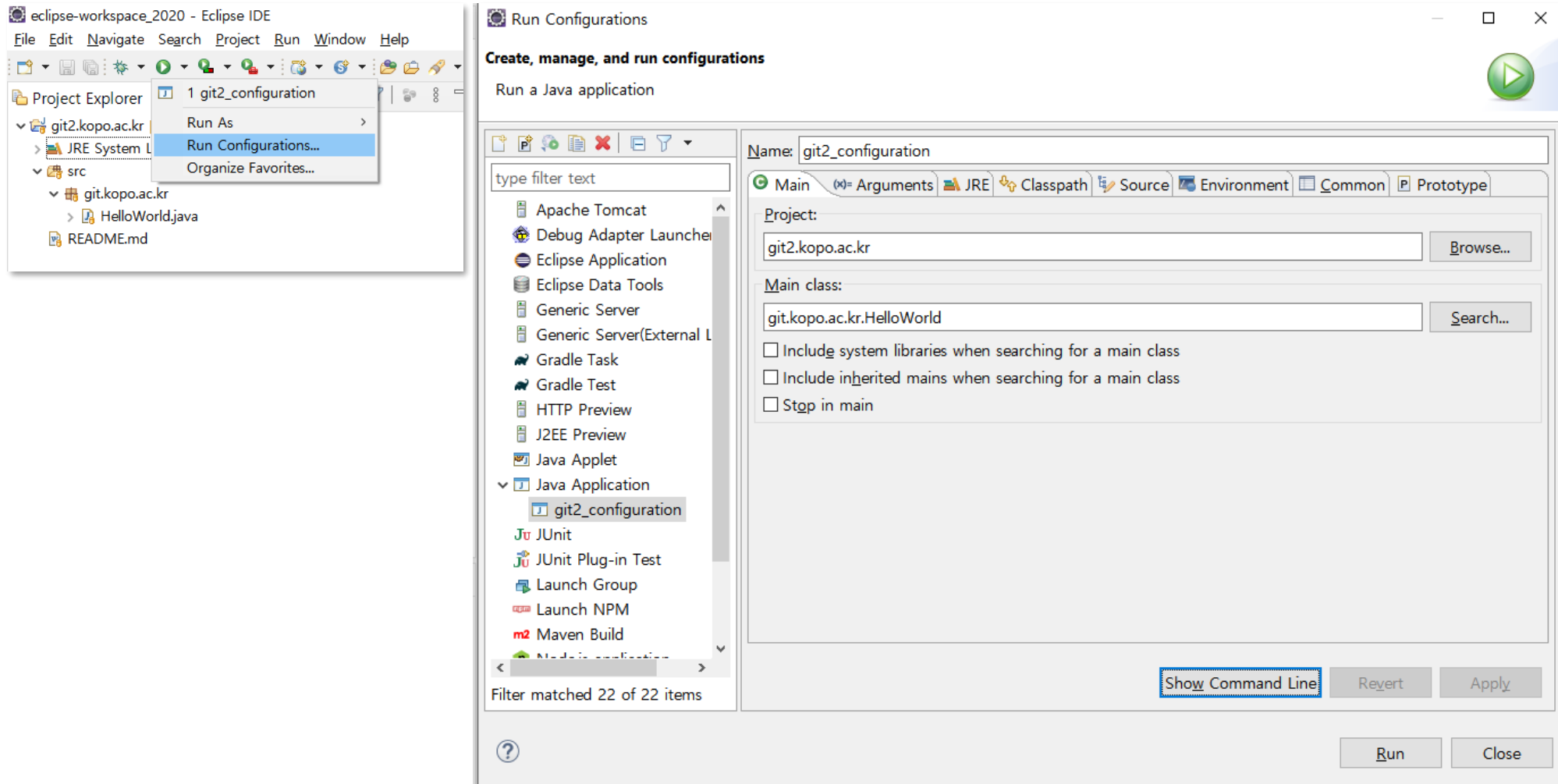
- Import Projects from File System of Archive 선택



- 완성!!

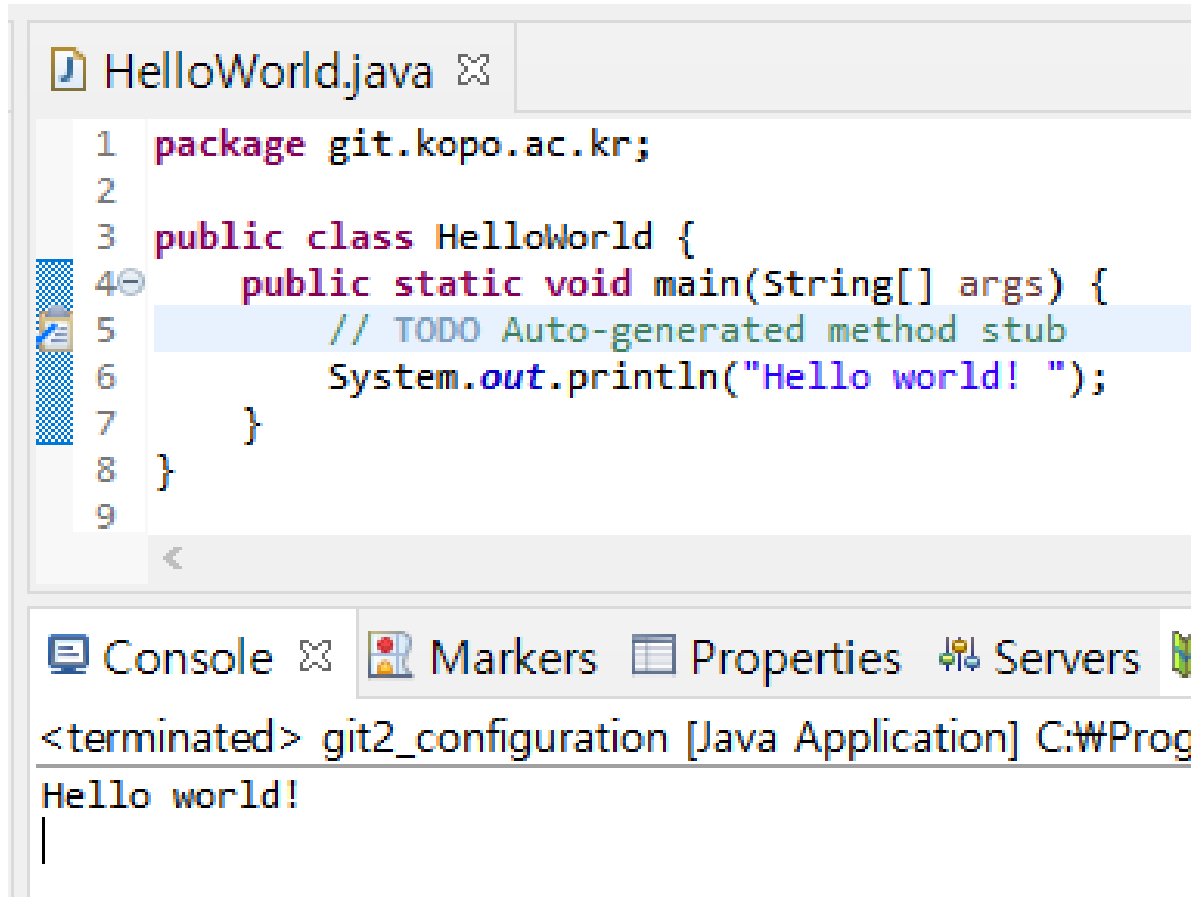
Clone한 repository를 실행시키자

- 실행 방법 정의 – Java Application으로 실행
 - Project와 main class도 명시



Clone한 repository를 실행시키자

- Hello World! 완성



The screenshot shows an IDE window with a tab for `HelloWorld.java`. The code is as follows:

```
1 package git.kopo.ac.kr;
2
3 public class HelloWorld {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         System.out.println("Hello world! ");
7     }
8 }
9
```

Below the code editor, there are tabs for `Console`, `Markers`, `Properties`, and `Servers`. The `Console` tab is active, displaying the output of the program:

```
<terminated> git2_configuration [Java Application] C:\WProg
Hello world!
|
```

Commit 하지 말아야 할 것들 설정

- Commit 하지 말아야 할 것들

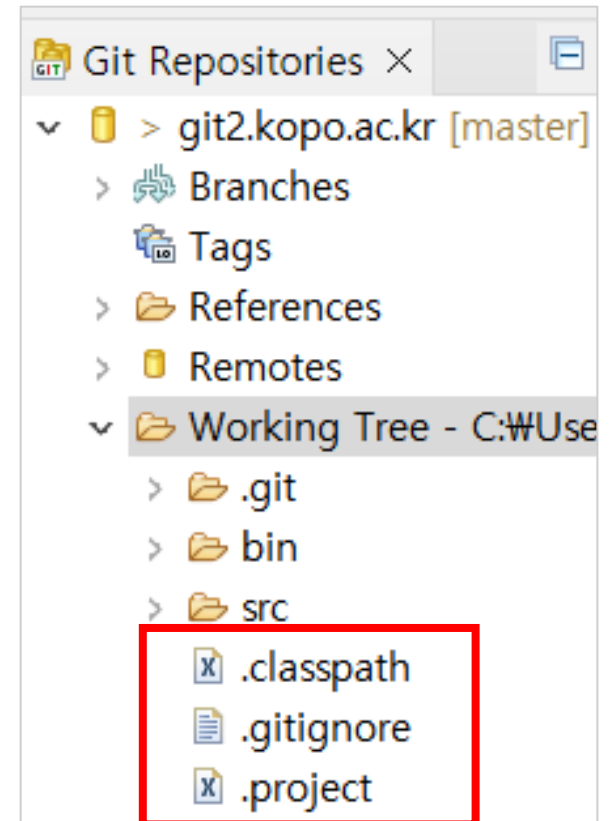
- eclipse 설정
- local 컴퓨터의 설정
- 보안과 관련된 정보
- 불필요한 것들
- 너무 크기가 큰 것들

- Commit 하지 않는 file들은 .gitignore 를 열고 설정

- .gitignore는 나중에도 계속 사용하므로 commit 하자

gitignore 설정

```
/bin/  
/.settings/  
.classpath  
.project
```





To do list

실습 1

- 옆친구 github repository를 clone해보자.
 - 1. 내 github로 fork를 해온다.
 - 2. local computer로 해당 repository를 clone한다.
 - 3. java 프로젝트를 실행해 본다.
-
- <https://github.com/heejinlee-kopo/git2.kopo.ac.kr.git>

실습 2

- 스스로 간단한 java application 실행과 git 연동까지 다시 해본다.
 - 교재를 보지 않고 실행해 볼 것.
 - 간단한 Java application 실행
 - 새로운 비어있는 github repository 생성
 - .gitignore 설정하기
 - 연결하고 commit and push하기
 - pull, push
- Git manual 읽기
- Git console 사용하기

아직 해보지 않은 것

- GUI tool을 활용해서 clone, branch생성, 삭제, checkout, commit, push, pull 등 해보기
 - Sourcetree app 활용해서 해보기
- 여러명 함께 사용할 때 벌어지는 상황들

과제

- **Sourcetree app을 활용하여 Git 기본 사용 방법 동영상 작성**

- <https://www.sourcetreeapp.com/>
- branch 생성, 삭제, checkout, commit, clone, push, pull 진행
- 해당 과정을 동영상으로 제작하여 본인 유튜브에 업로드
- Git 과목 페이지에 sourcetree 설명 동영상을 보여줄 것

- **Git 수업 홈페이지 만들기**

- Git 수업에 대한 개요
- 과제 github 시작하기 동영상
- 과제 sourcetree 사용 동영상
- 힘들지만 극복한점

- **제출 방법**

- Git 수업 홈페이지 url제출 - 예) <https://{본인id}.github.io/{repository명}>
- 본인 유튜브 채널에 sourcetree app활용 동영상 업로드 및 url 제출
 - 제목: SourceTree활용한 git 사용법 한국폴리텍대학 광명융합기술교육원 데이터분석과 {본인 학번} 홍길동
- OneDrive 2. 소프트웨어 형상관리/과제제출/과제제출리스트.xml 에 sheet2 과제2_SourceTree활용에 git 홈페이지, YouTube url 기재
- OneDrive 2. 소프트웨어 형상관리/ 과제제출/과제2_SourceTree활용하기 폴더에 활용 동영상 upload

- **제출 기한**



Thank you