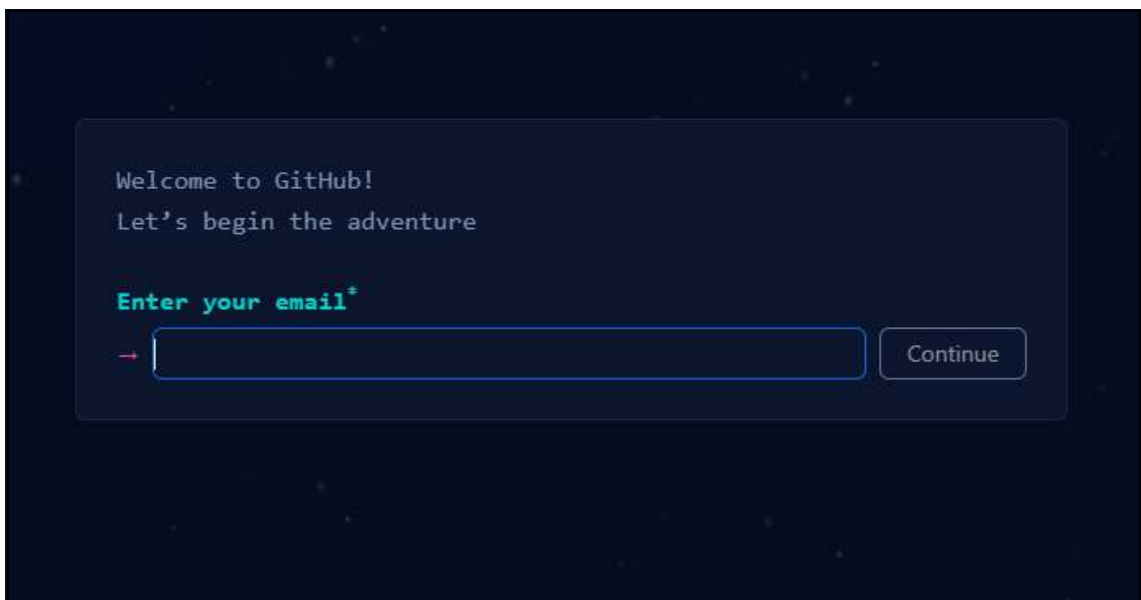
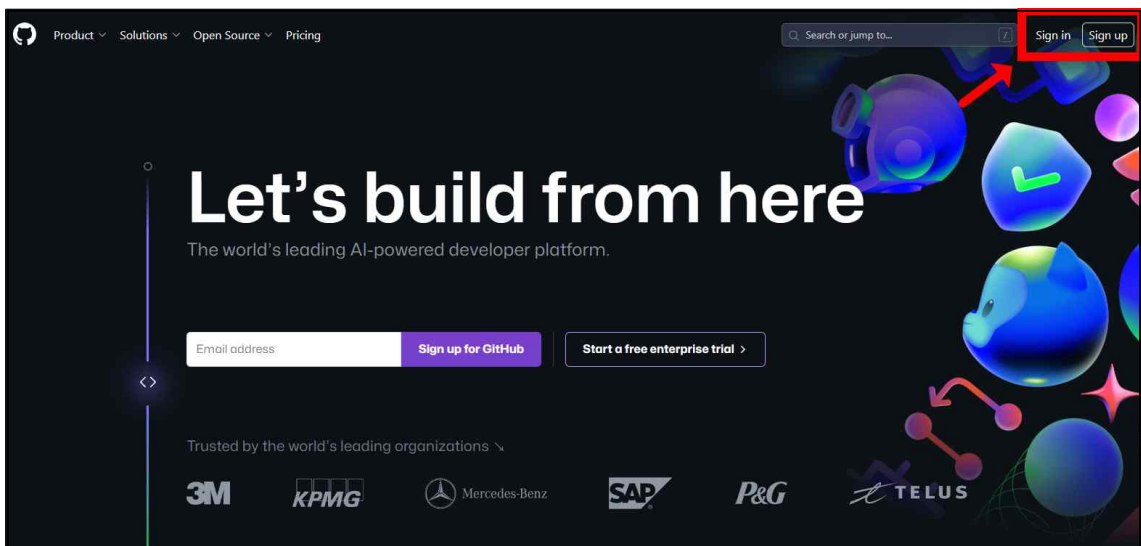
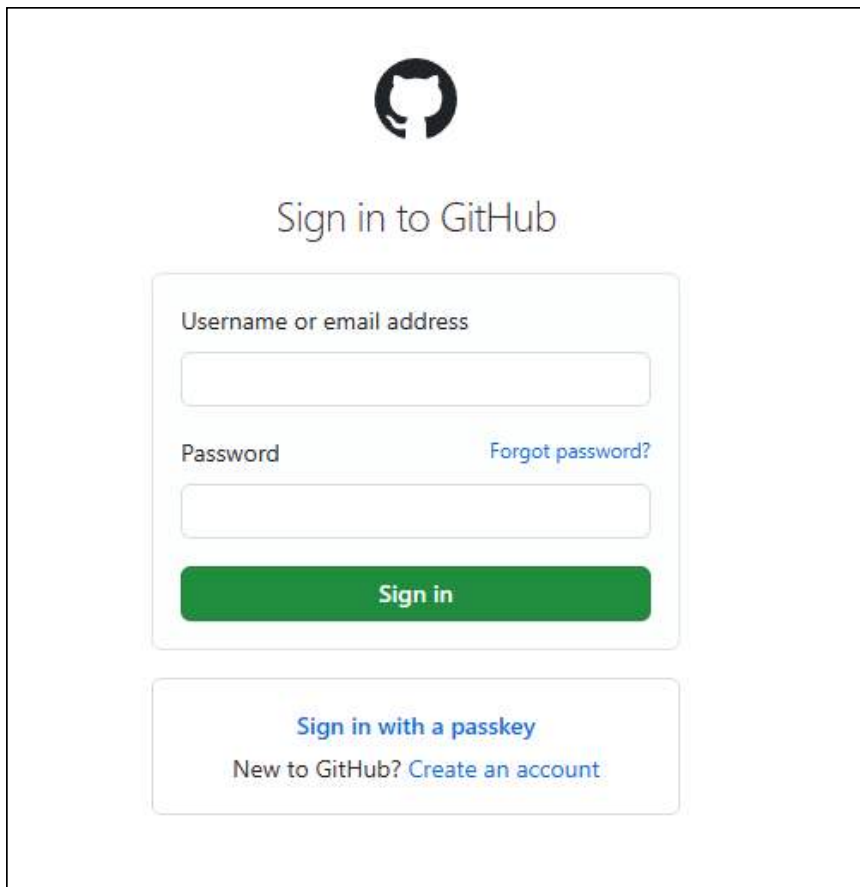


GitHub 와 이클립스 연동 사용

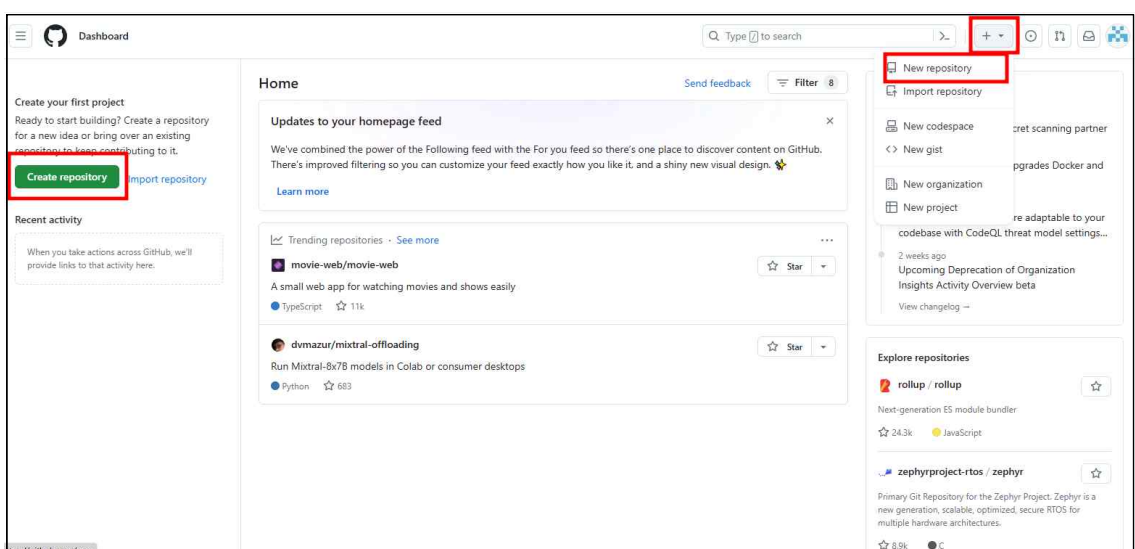
1. GitHub에 저장소 생성 & 토큰키 생성(팀원 중 1명만 실행)

- GitHub(<https://github.com>)에 가입(Sign up) 및 로그인(Sign in)하고 Repository를 생성합니다





- 로그인(Sign in)된 화면의 왼쪽상단이나 오른쪽 상단의 + 메뉴 옆의 펼침으로 Create repository 나, New repository를 선택합니다



- Repository name을 입력합니다. 특별히 보안이 설정된 프로젝트

트가 아니라면 Public 선택사항을 그대로 둡니다. 파일 업로드 상태를 최초에 확인하려면 Add a README file 도 선택합니다

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

Repository name *

heejoonk

/ ShoppingMall

ShoppingMall is available.

Great repository names are short and memorable. Need inspiration? How about [redesigned-spork](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

- 그리고 최종 화면 하단의 Create repository를 눌러서 Repository를 생성합니다

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

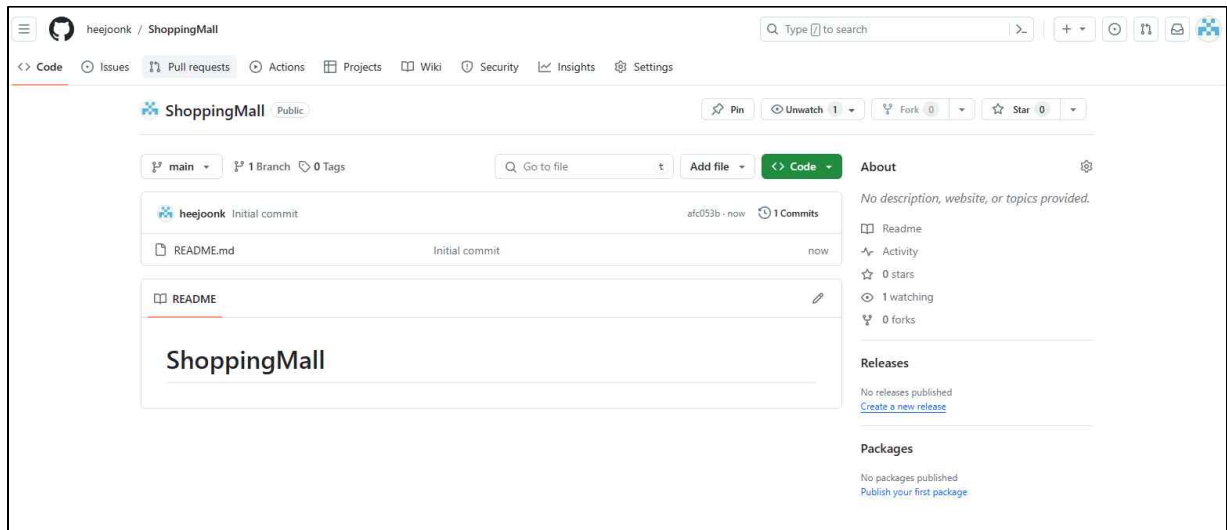
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

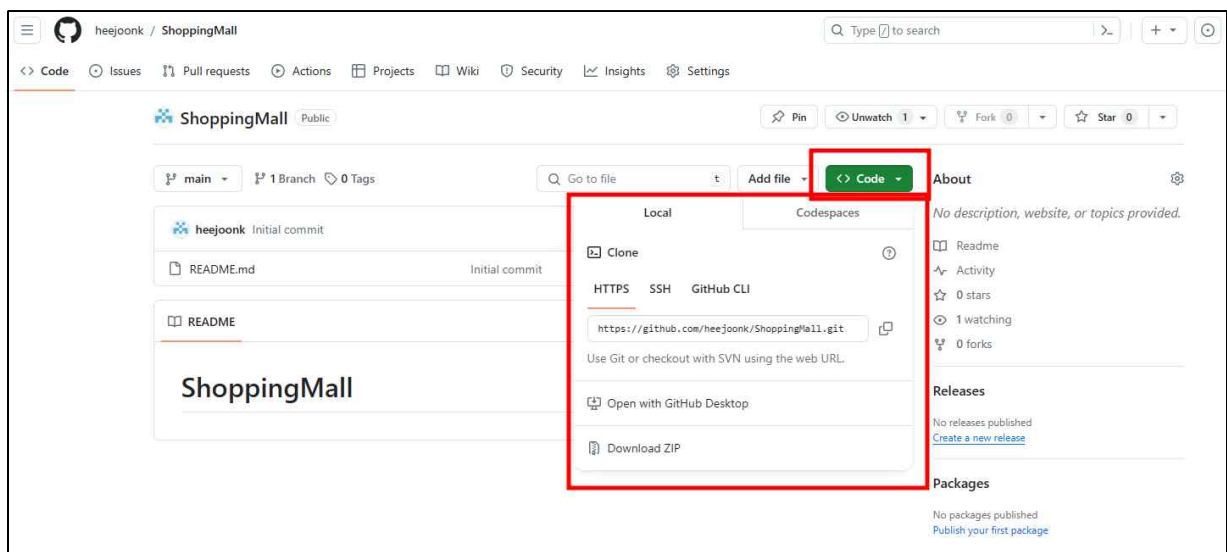
You are creating a public repository in your personal account.

Create repository

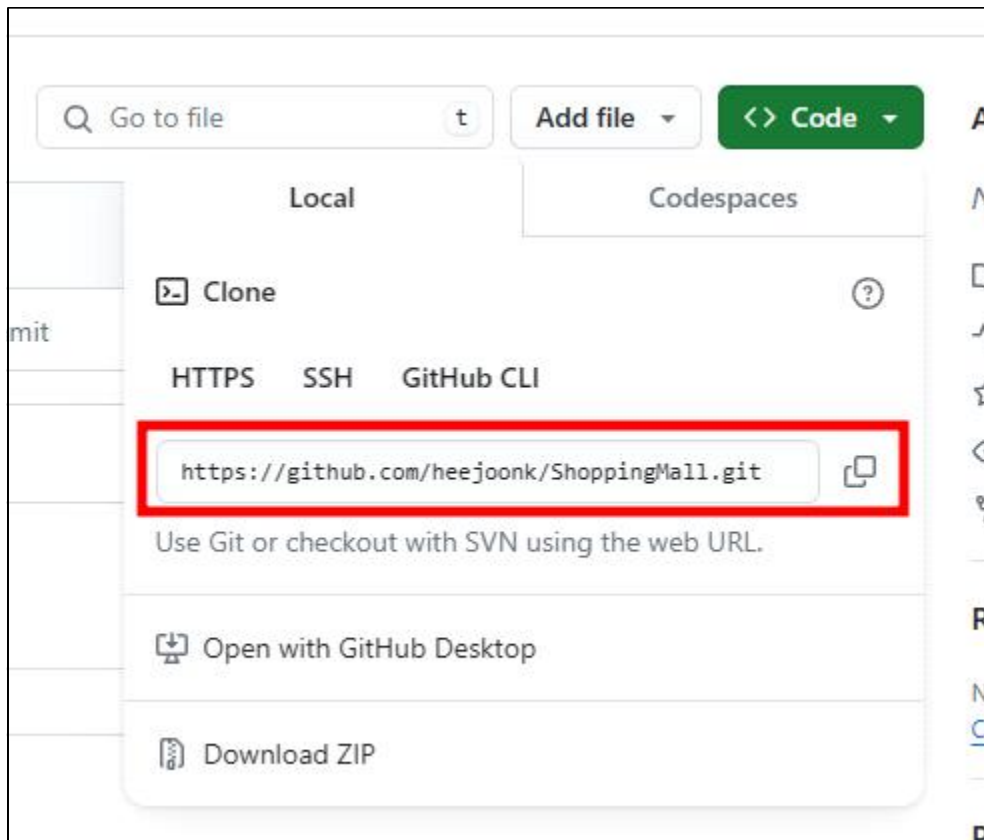
■ Repository 가 생성된 모습입니다.



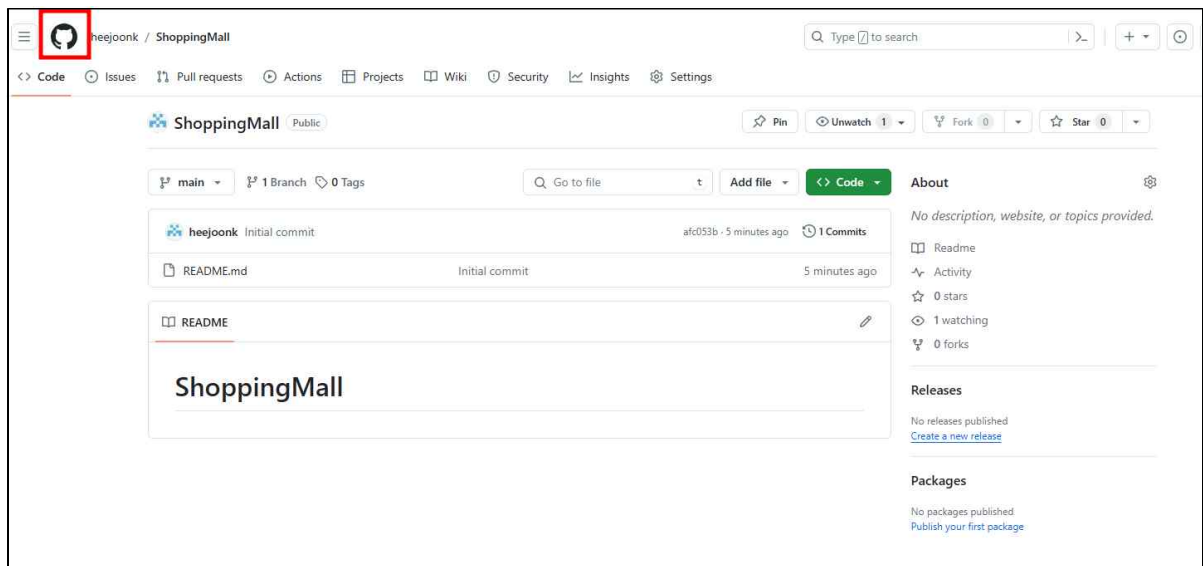
■ 오른쪽 상단부의Code를 눌러서 나오는 펼침메뉴를 확인합니다



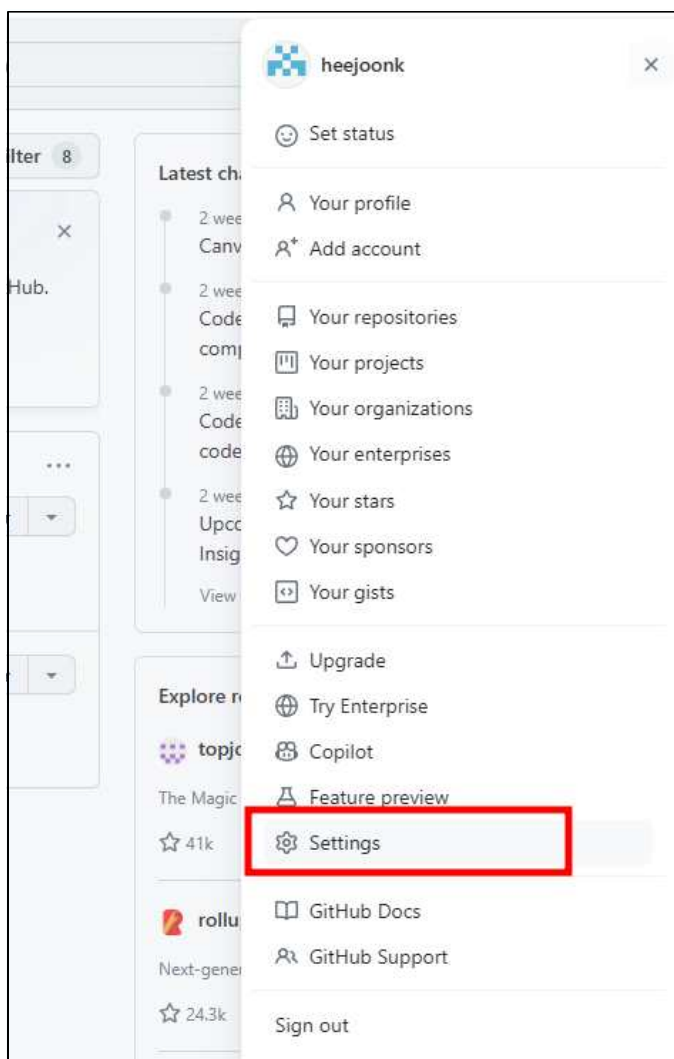
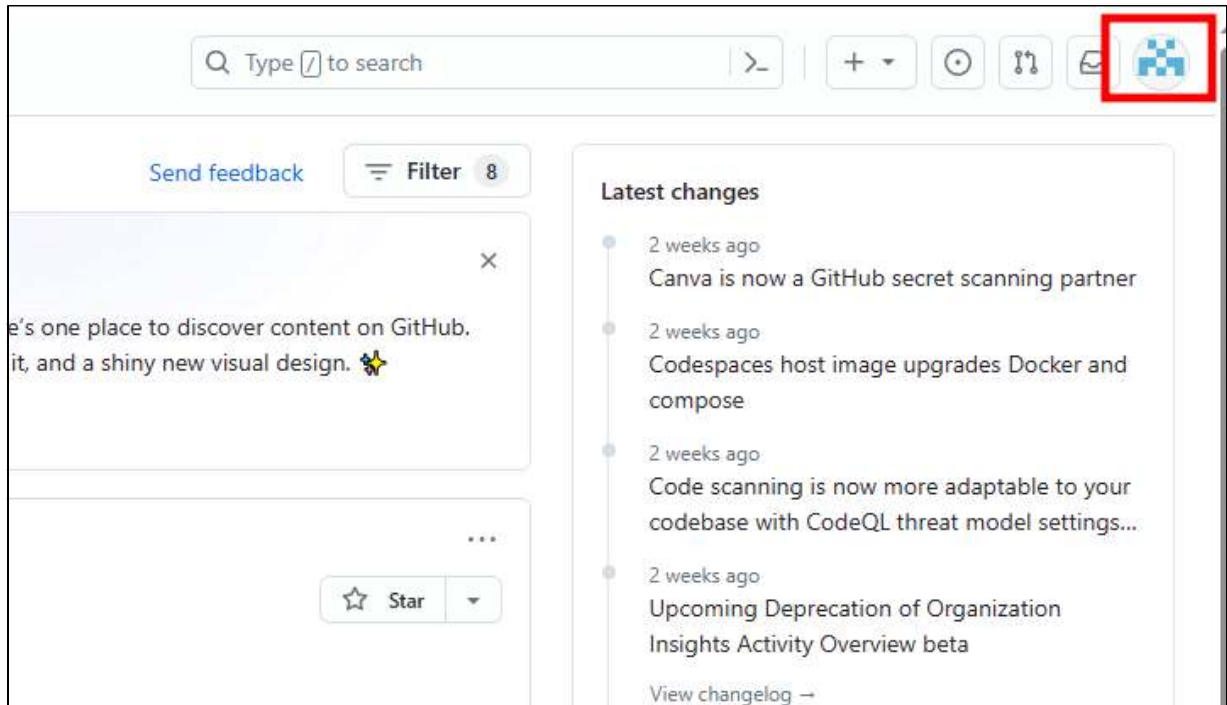
■ 아래 화면에 표시된 주소를 복사해서 팀원들에게 배포합니다. 모든 자료들이 공유될 주소 URL입니다



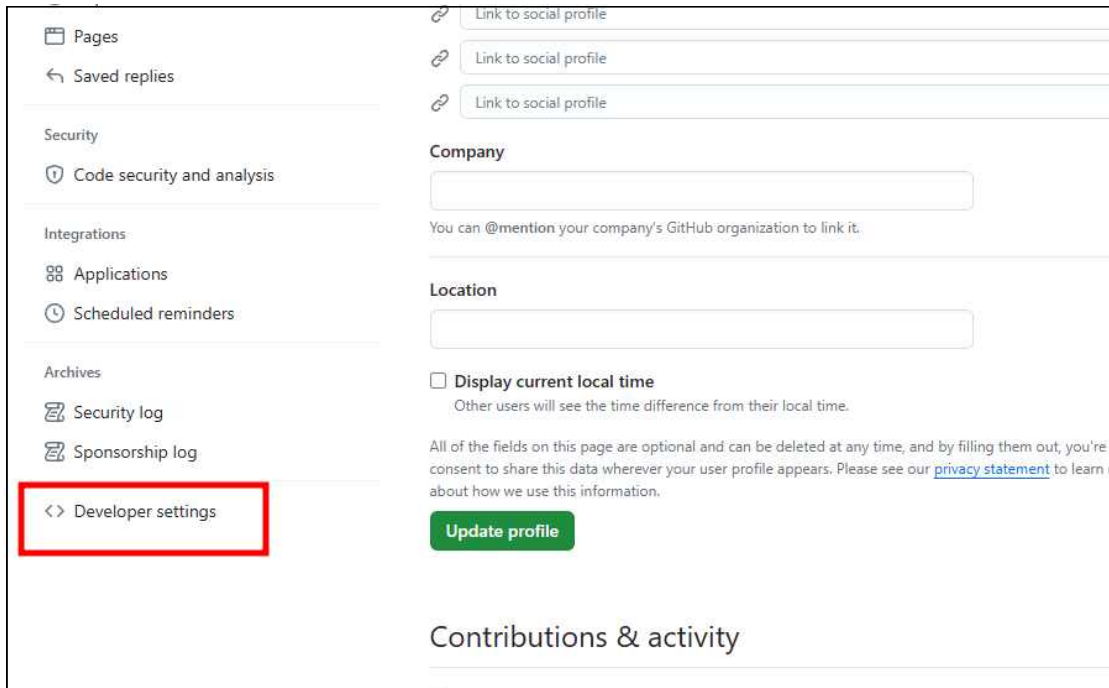
- 이제 이 Repository 에 접근할 때 사용할 토큰키를 발급 받습니다 (팀원들의 접근 방법이 아이디/비번 방식에서 토큰키방식으로 몇 년전에 변경되었습니다). 왼쪽 상단의 아이콘을 클릭해서 메인화면으로 갑니다



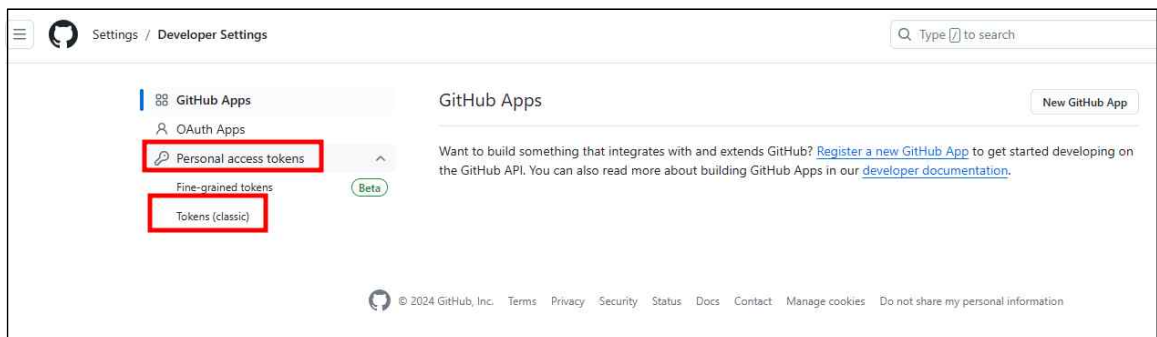
- 오른쪽 상단의 개인 계정 아이콘을 누르고 setting 메뉴를 선택합니다.



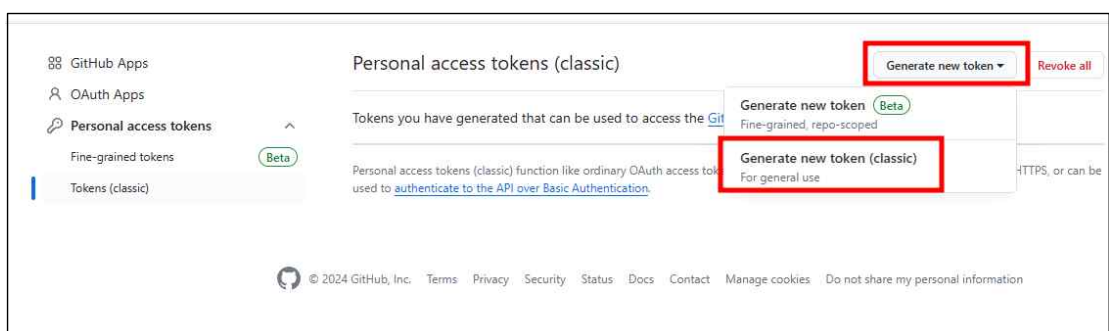
- 이동한 화면의 왼쪽 맨하단의 Developer settings를 클릭합니다



- 왼쪽 메뉴의 Personal access tokens를 누르고 메뉴가 펼쳐지면 Tokens(classic)을 선택합니다



- 표시된 화면에서 오른쪽 상단의 Generate new token을 클릭하고 펼쳐진 메뉴에서 Generate new token(classic)을 다시한번 선택합니다



- 생성화면에서 Note를 입력하고 아래 선택사항들은 가능한 모두 선택합니다. Expiration(키의 유효기간)을 선택해주되 여기서는 보안상의 위험을 크게 다루지 않을것이므로 No expiration을 선택하겠습니다

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Shoppingmall

What's this token for?

Expiration *

No expiration

The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

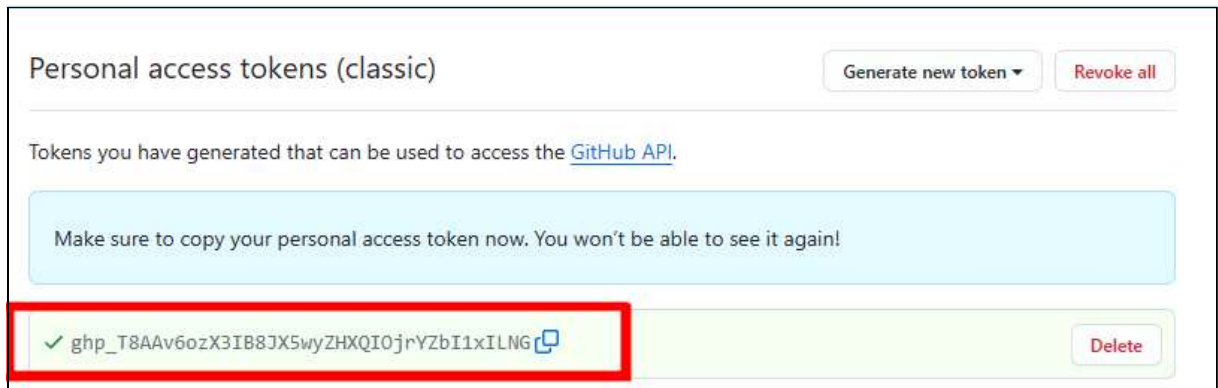
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry

- 마지막으로 하단의 Generate token을 눌러 토큰생성을 완료합니다

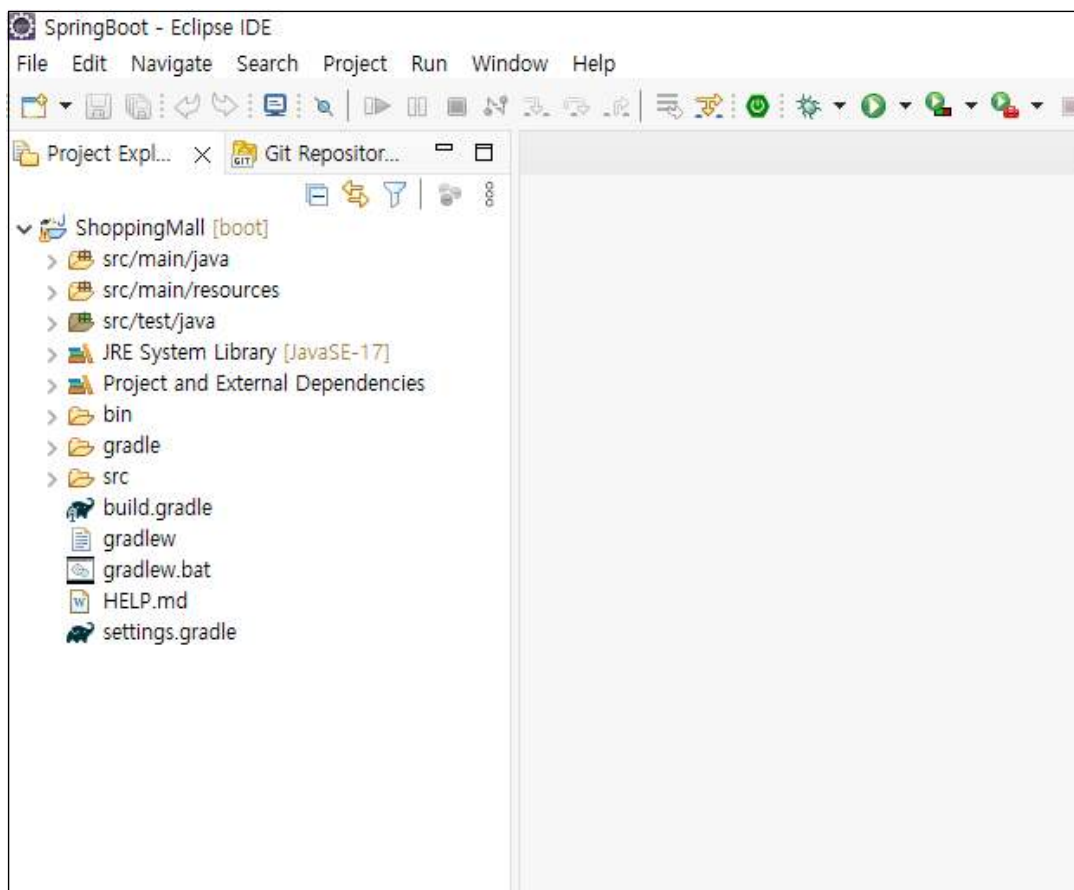
<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input checked="" type="checkbox"/> write:gpg_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:gpg_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input checked="" type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate tokenCancel

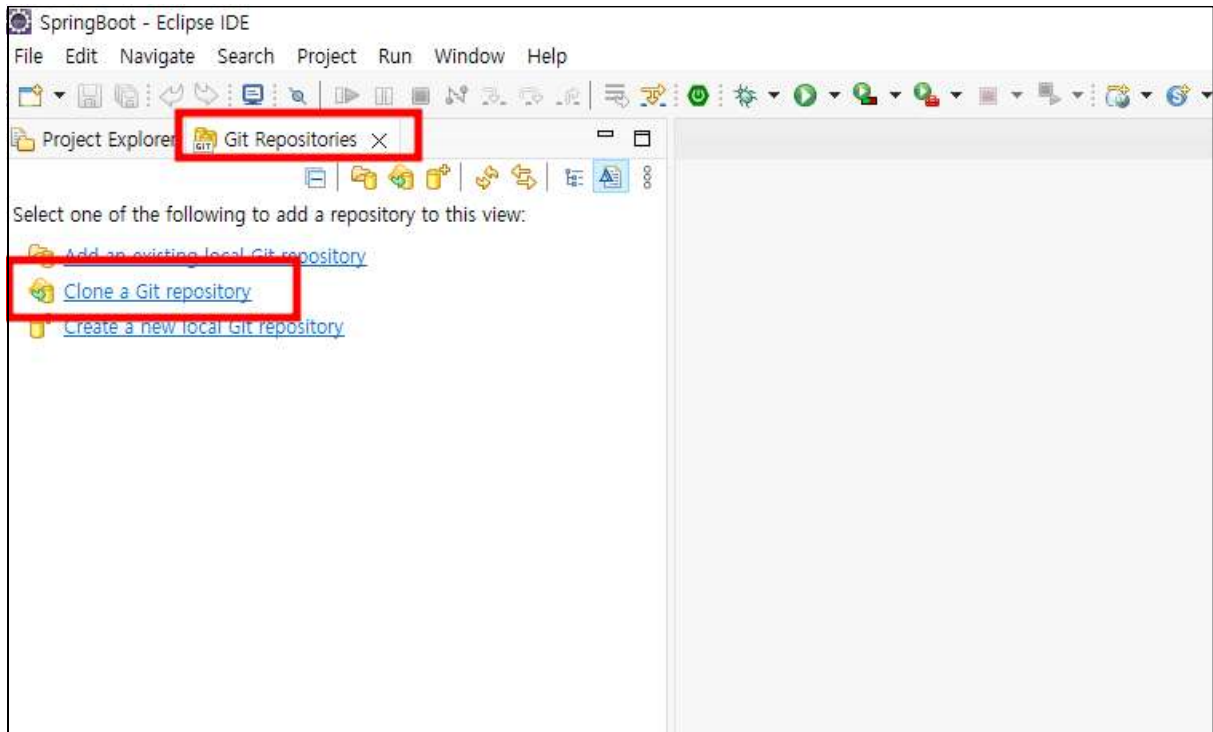
- 생성된 토큰을 확인하고 복사해서 팀원들에게 배포합니다. 이키가 없으면 Repository 에 접근(Push) 가 불가능합니다. 또한 아래 화면은 딱한번 표시되고 없어지므로 지금 복사해두지 않고 토큰키를 잃어버리면 재발급을 받아야 하니 반드시 복사해서 따로 메모하세요



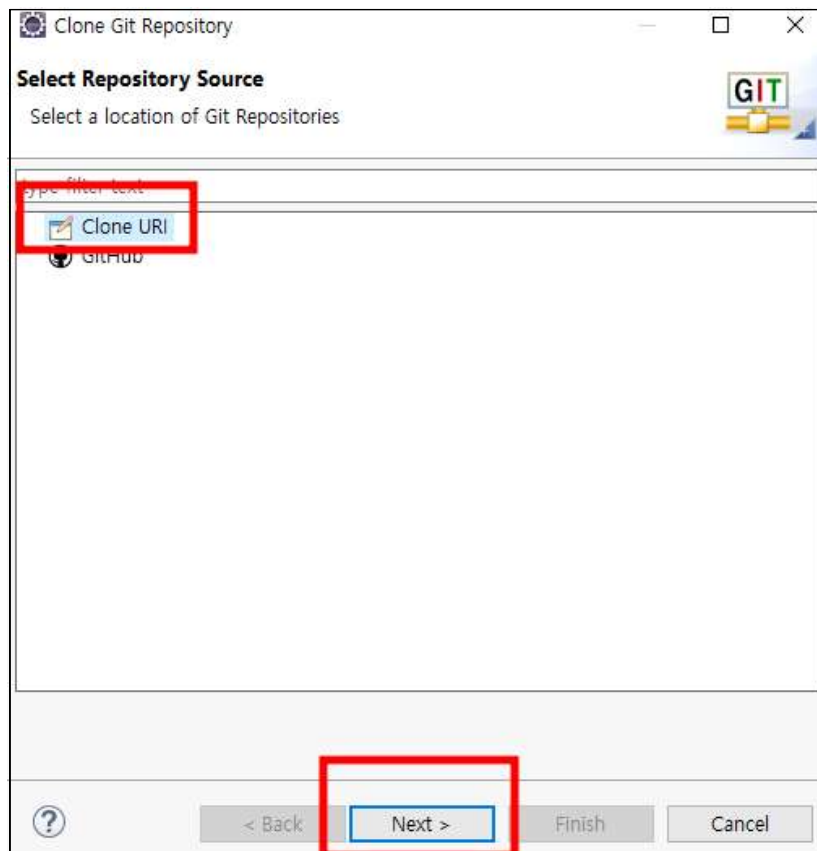
- 이제 이클립스에 새로운 프로젝트를 생성하고 GitHub의 Repository에 업로드(PUSH)하여 팀원들이 공유해서 사용할수 있게 합니다. 새로운 프로젝트를 만듭니다



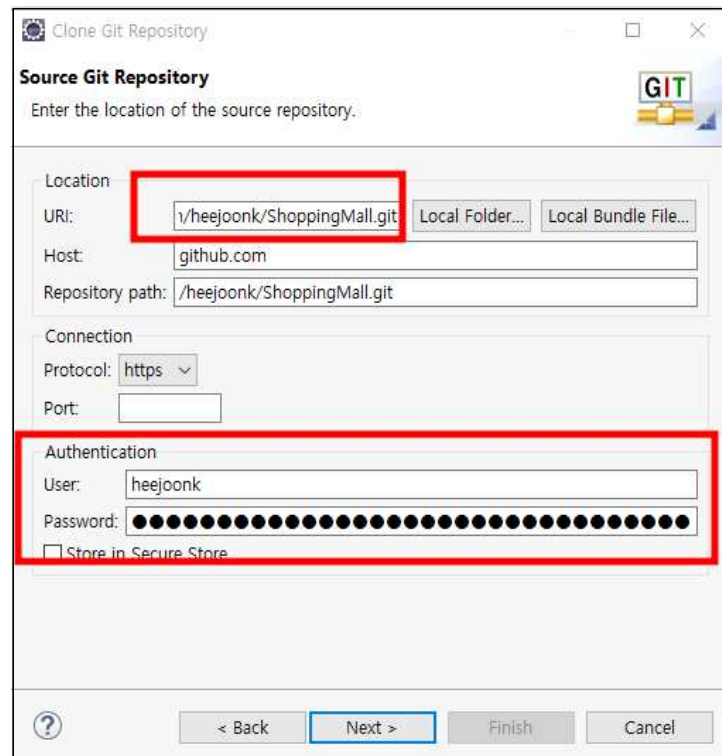
- Git Repository 창을 열어 Clone a Git repository를 선택합니다. 만일 Git Repository 창이 안보인다면 Window-Show view-other에서 git으로 검색하면 열수 있습니다.



- 표시된 화면에서 clone URI를 누르고 Next를 클릭합니다

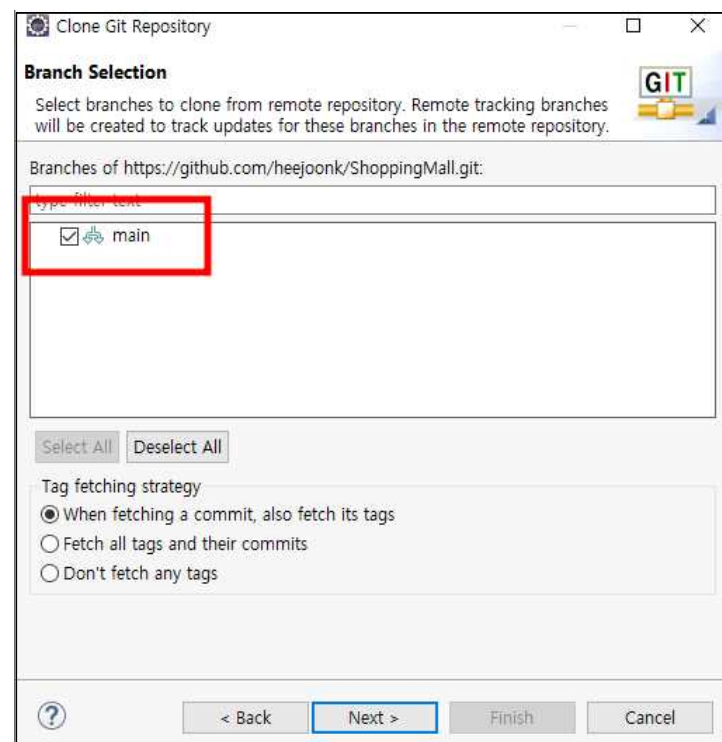


- 표시된 화면에서 앞단계에서 itHub 에 Repository를 생성하고 배포했던 URL을 입력합니다. 그리고 User 와 Password 는 현재 계정의 아이디 와 생성한 토큰키를 입력합니다



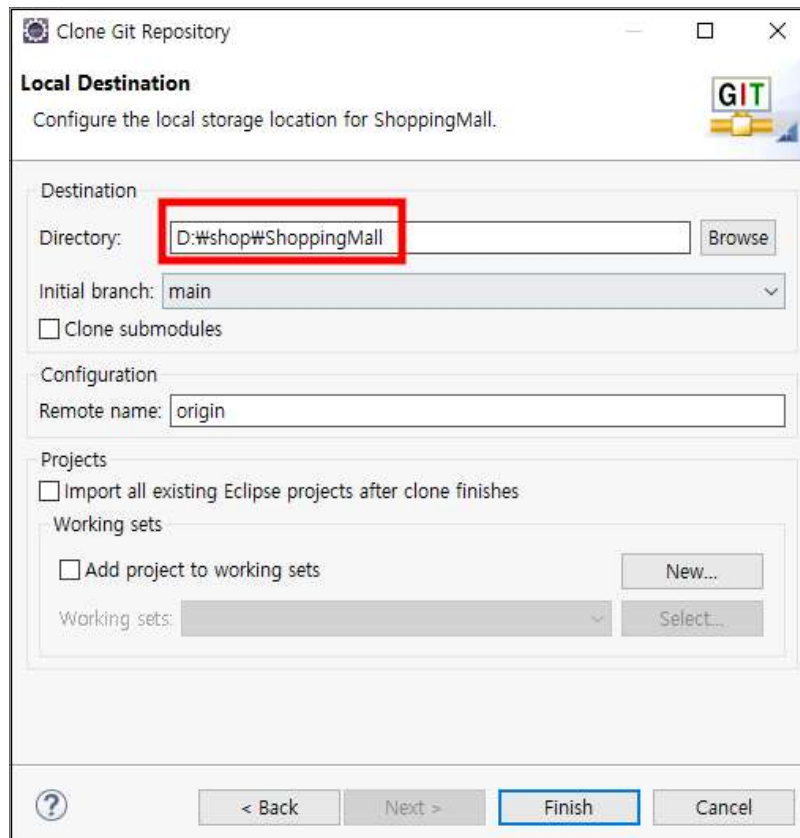
The image shows the 'Clone Git Repository' dialog box, specifically the 'Source Git Repository' step. The dialog has a title bar with a Git logo and standard window controls. Below the title, it says 'Source Git Repository' and 'Enter the location of the source repository.' The 'Location' section contains a 'URI:' field with the value 'v/heejoonk/ShoppingMall.git' (highlighted with a red box), a 'Host:' field with 'github.com', and a 'Repository path:' field with '/heejoonk/ShoppingMall.git'. There are buttons for 'Local Folder...' and 'Local Bundle File...'. The 'Connection' section has a 'Protocol:' dropdown set to 'https' and an empty 'Port:' field. The 'Authentication' section (highlighted with a red box) has a 'User:' field with 'heejoonk' and a 'Password:' field filled with black dots. There is a checkbox for 'Store in Secure Store' which is unchecked. At the bottom, there are buttons for '< Back', 'Next >' (highlighted with a blue box), 'Finish', and 'Cancel'.

- 다음 화면에서 main을 클릭하고 선택확인 후 Next를 클릭합니다

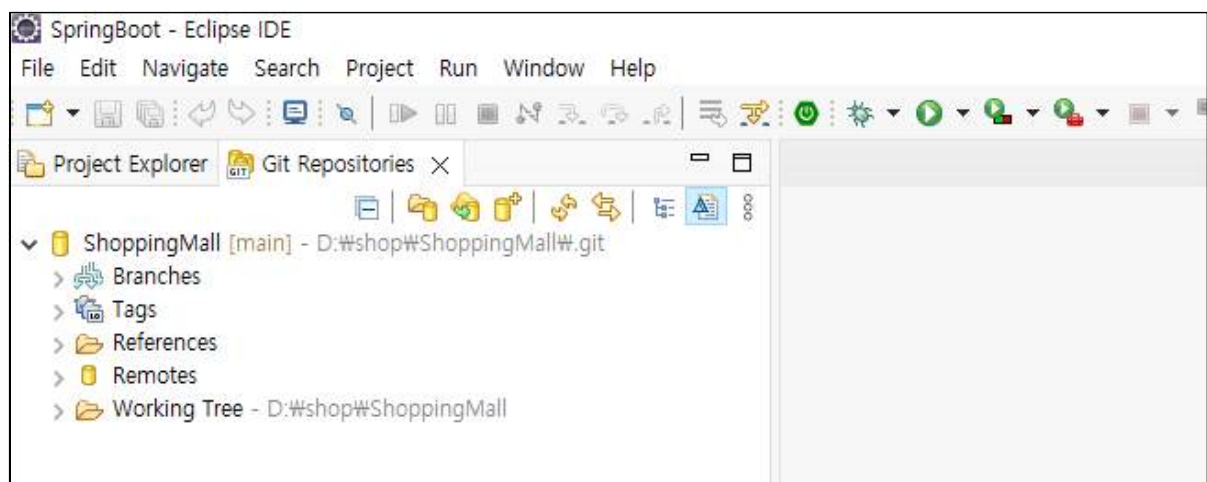


The image shows the 'Clone Git Repository' dialog box, specifically the 'Branch Selection' step. The dialog has a title bar with a Git logo and standard window controls. Below the title, it says 'Branch Selection' and 'Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.' The 'Branches of https://github.com/heejoonk/ShoppingMall.git:' section has a search filter and a list of branches. The 'main' branch is selected with a checkbox (highlighted with a red box). Below the list are 'Select All' and 'Deselect All' buttons. The 'Tag fetching strategy' section has three radio buttons: 'When fetching a commit, also fetch its tags' (selected), 'Fetch all tags and their commits', and 'Don't fetch any tags'. At the bottom, there are buttons for '< Back', 'Next >' (highlighted with a blue box), 'Finish', and 'Cancel'.

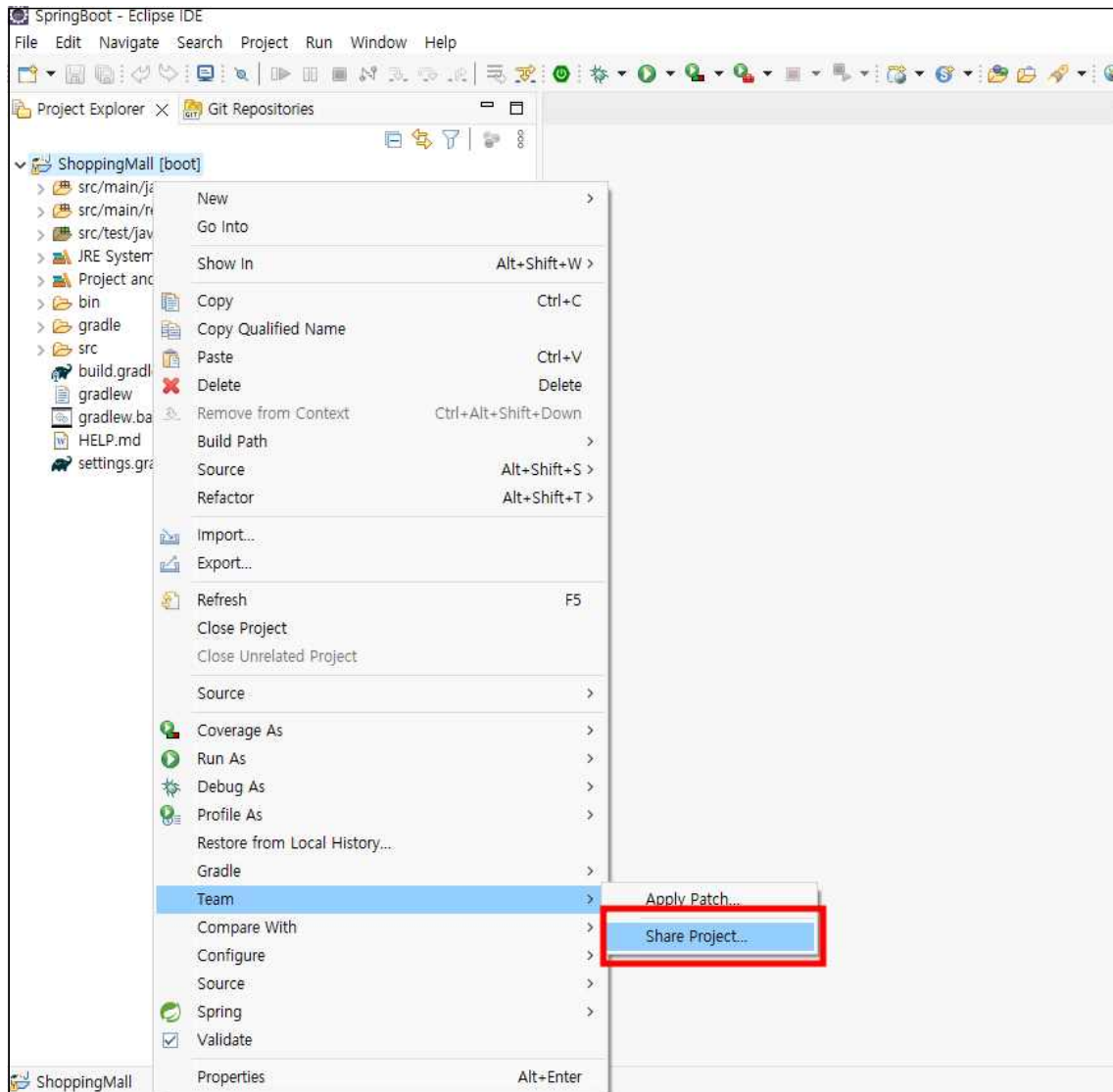
- 다음에 표시되는 화면에서는 GitHub의 저장장소와 현재 내컴퓨터와 연결될 공유 폴더를 설정합니다. 이는 나의 프로젝트 파일과 GitHub 사이에 위치하는 공유폴더로 변화되는 내용에대한 완충역할을 할 장소이므로 별도의 폴더를 만들어서 설정합니다. 여기서는 D드라이브에 shop 폴더를 만들어서 설정합니다



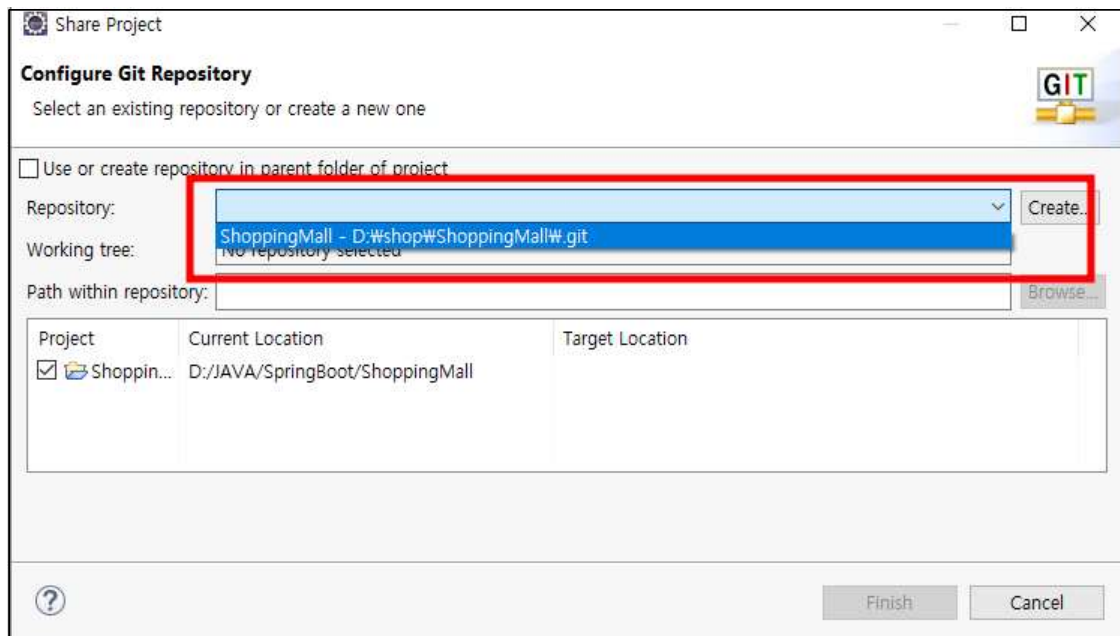
- finish를 누르면 왼쪽과 같이 연결된 Repository 가 표시되고 이를 local Repository라고도 부릅니다



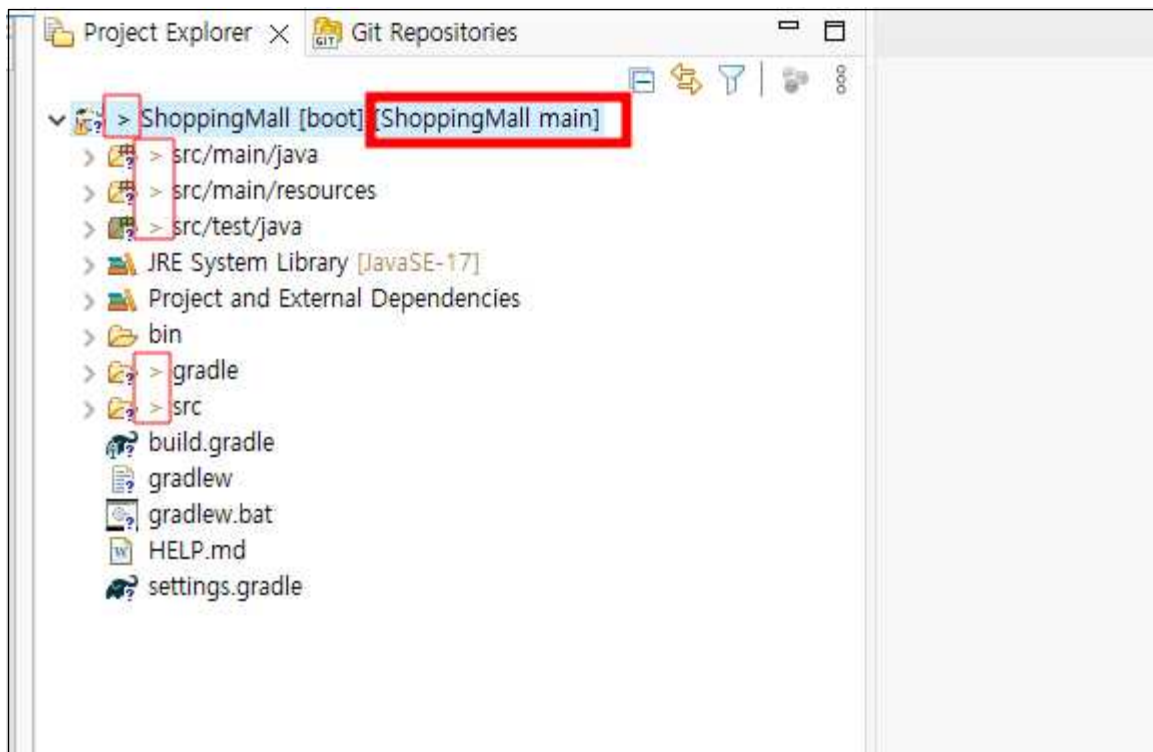
- 그리고 이제 Project Explorer에서 새로 만든 프로젝트를 이 로컬 Repository에 연결합니다. 프로젝트 이름 위에서 마우스 오른쪽버튼을 클릭하고 Team 메뉴에서 Share Proejct를 선택합니다



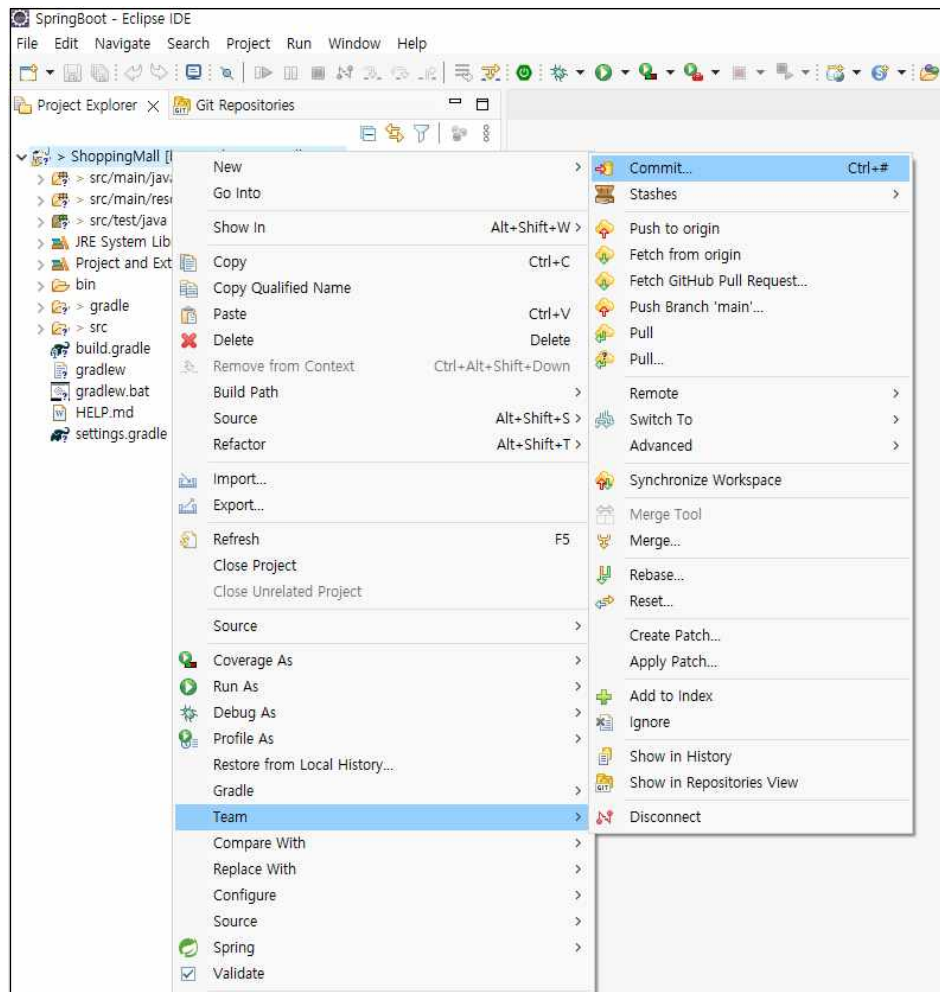
- 표시된 화면에서 Reposotry 항목을 클릭하면 방금 만든 로컬 Repository가 펼쳐져 선택할 수 있습니다 선택하고 finish를 누릅니다



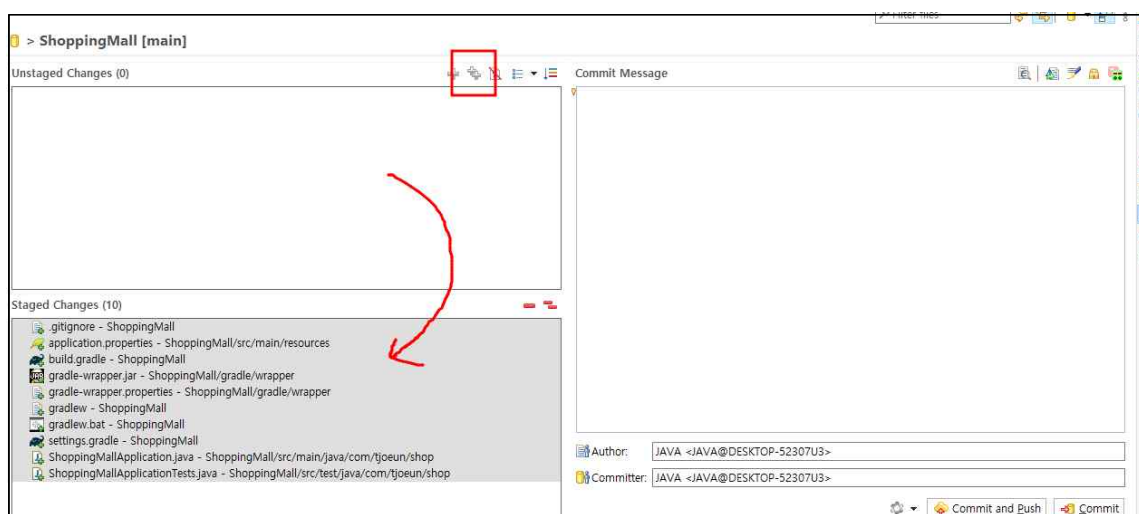
- 공유가 잘 되었다면 프로젝트 이름옆에 local Repository 이름이 붙었을꺼고 실제 GitHub 의 내용과 다른 파일과 폴더에 꺾쇠괄호표시가 붙습니다 이들이 최초 업로드될 파일들입니다.



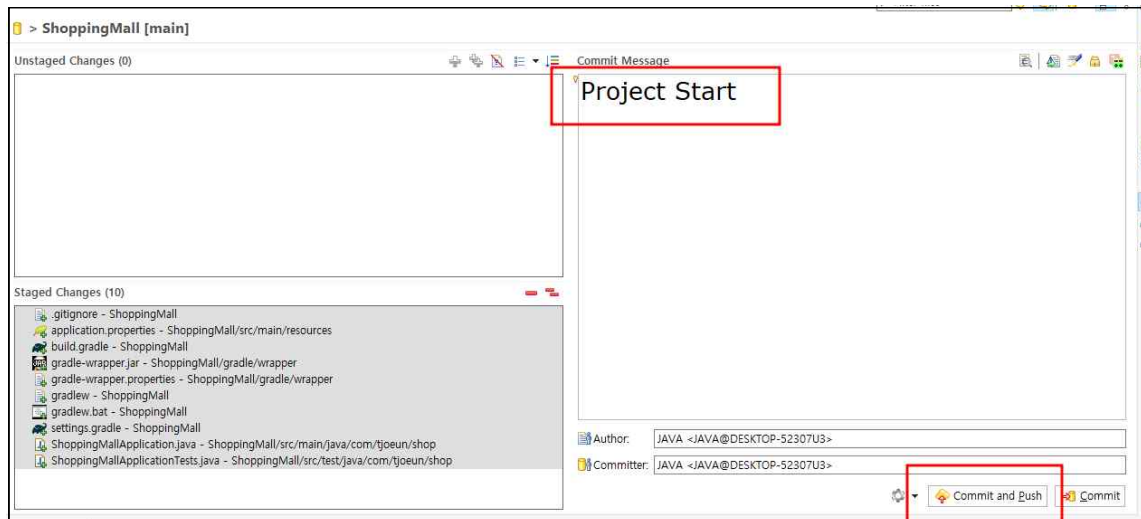
- 이제 GitHub에 프로젝트를 업로드(PUSH)합니다. 프로젝트 이름 위에서 오른쪽 버튼을 누르고 Team 메뉴에 가면 이제 메뉴가 많아졌습니다. 거기서 Commit을 선택합니다



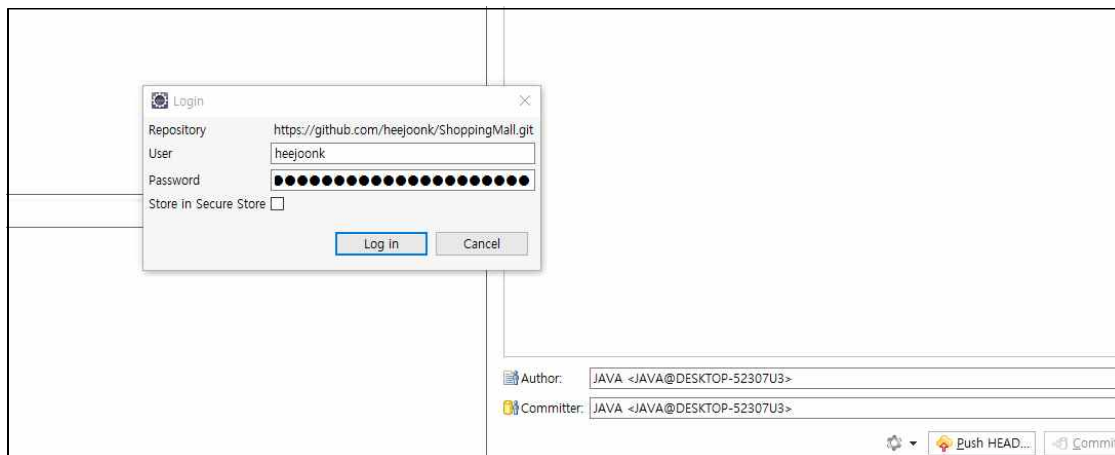
- Git Staging 화면이 나오면 왼쪽 Unstaged Chanes 의 내용을 표시된 버튼을 눌러서 아래 Staged Changes로 옮깁니다



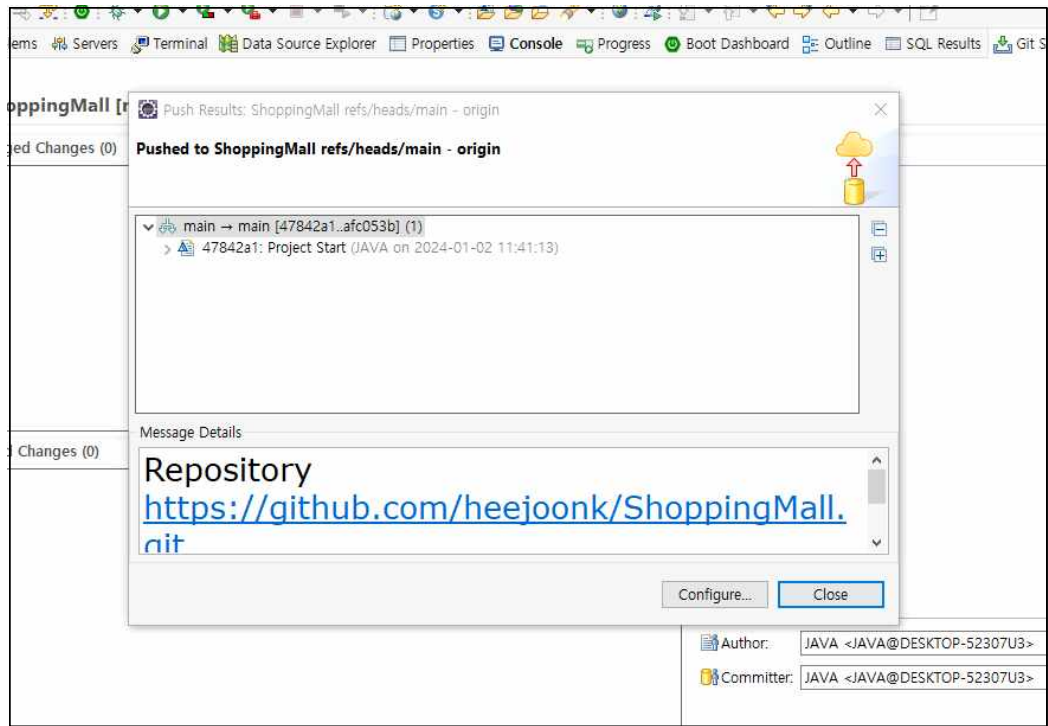
- 오른쪽 Commit 메시지에 Project Start를 입력하고 Commit and Push를 클릭합니다



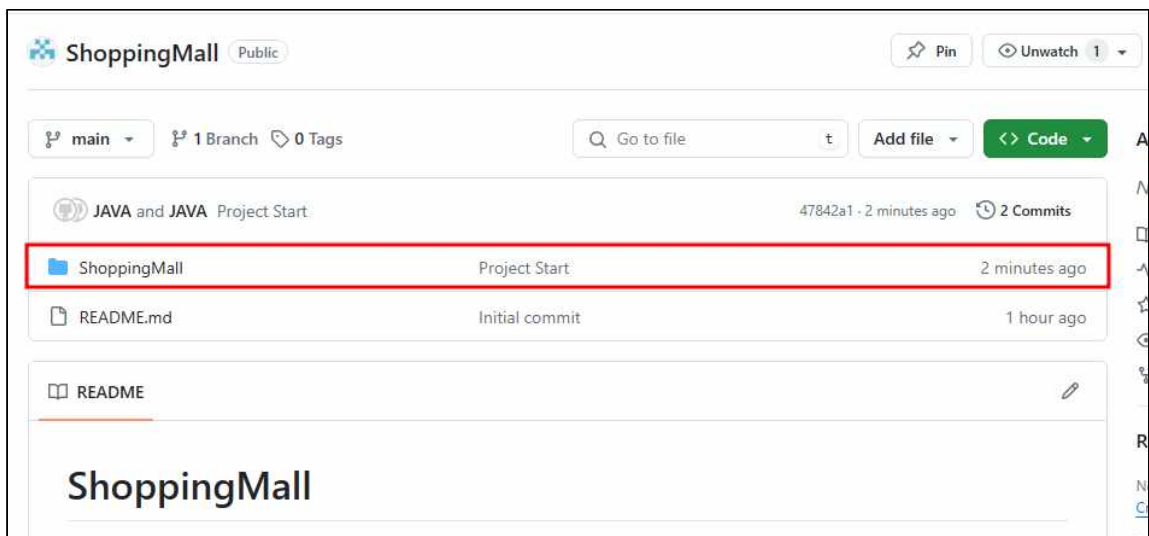
- 이후 나오는 Login 창은 Userid와 토큰키를 입력합니다



- 아래와 같은 화면이 나오면 정상 업로드 된 상태입니다

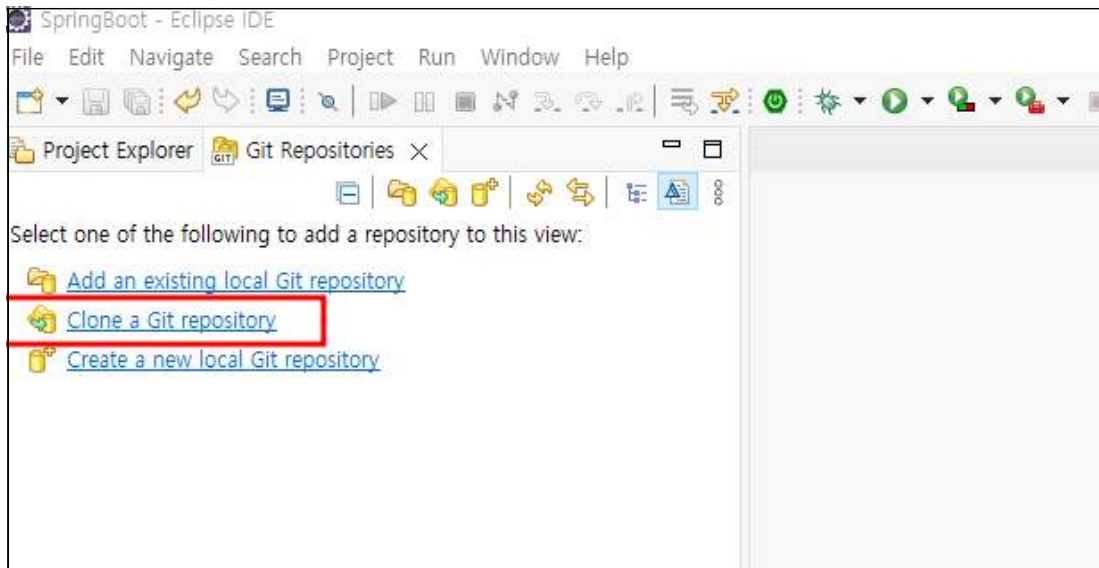


- GitHub 사이트에서 새로고침을 하면 현 프로젝트 폴더가 업로드된 화면을 확인할 수 있습니다

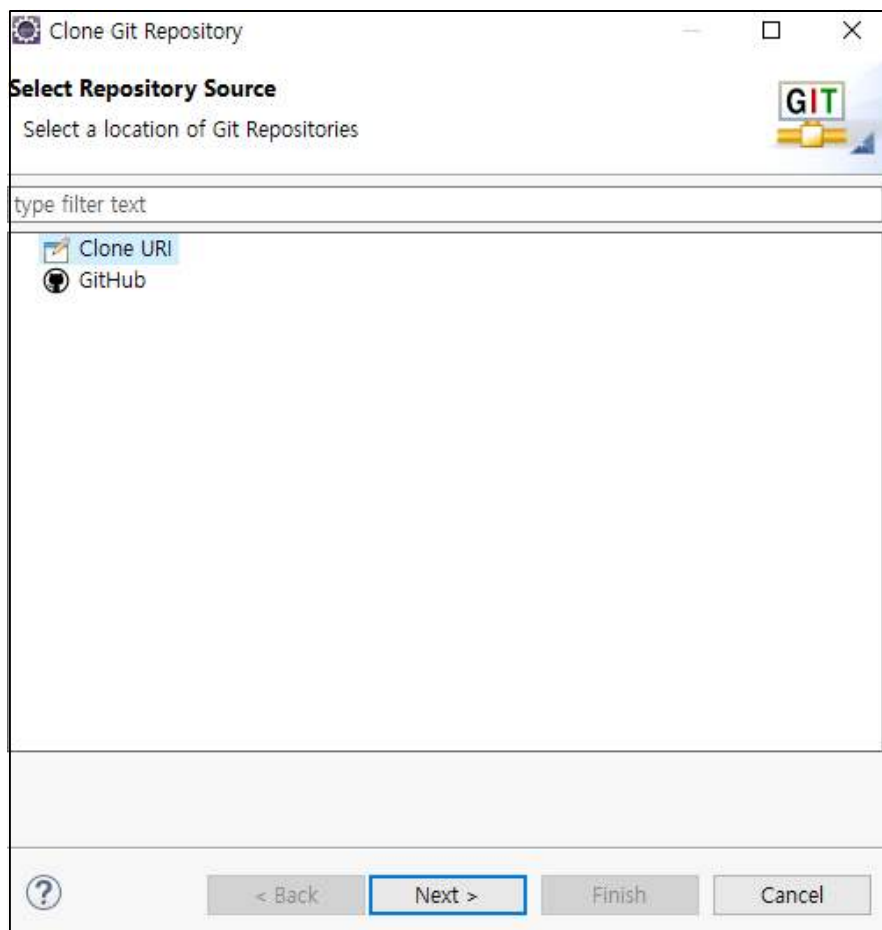


2. GitHub의 Proejct.를 이클립스로 다운로드(PULL) - 팀원 모두 실행

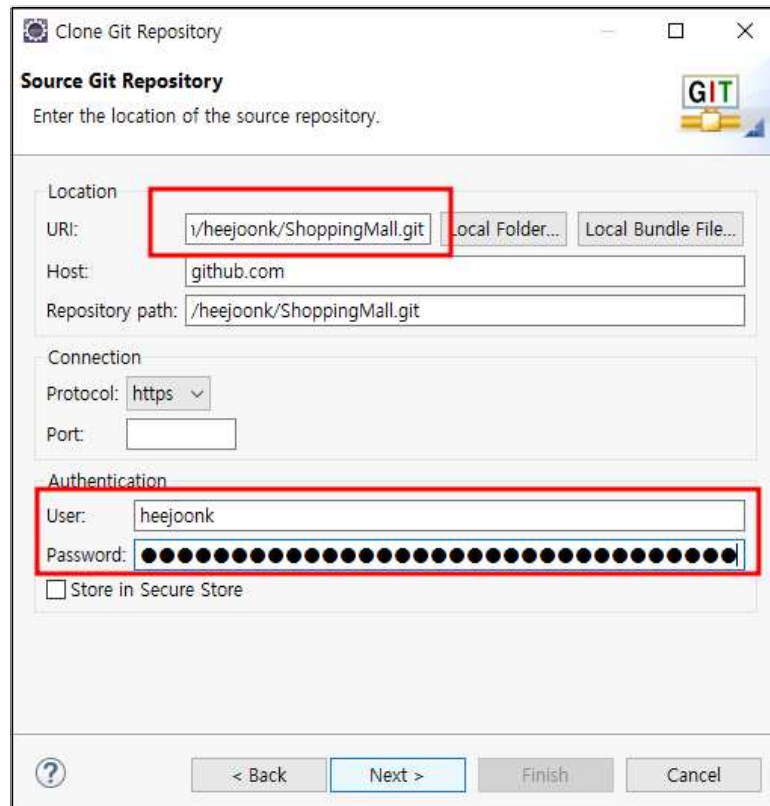
■ Git Repository 창에서 Clone a Git repository를 선택합니다



■ Clone URI를 선택하고 Next를 클릭합니다

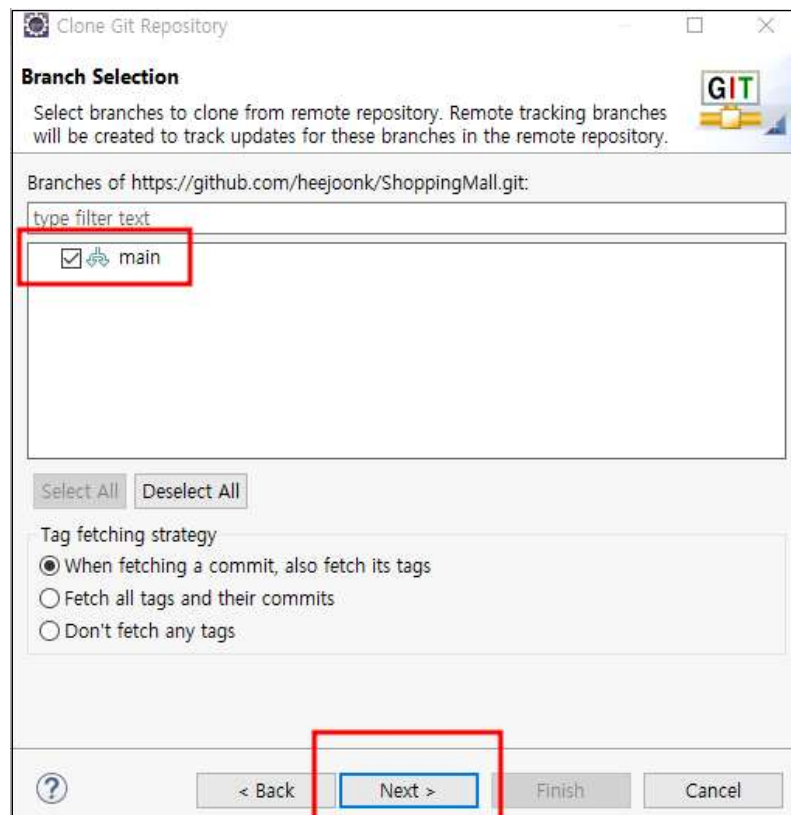


- 배포받은 Repository URI를 입력합니다. 그리고 User 와 Password 는 아이디와 토큰키를 입력합니다



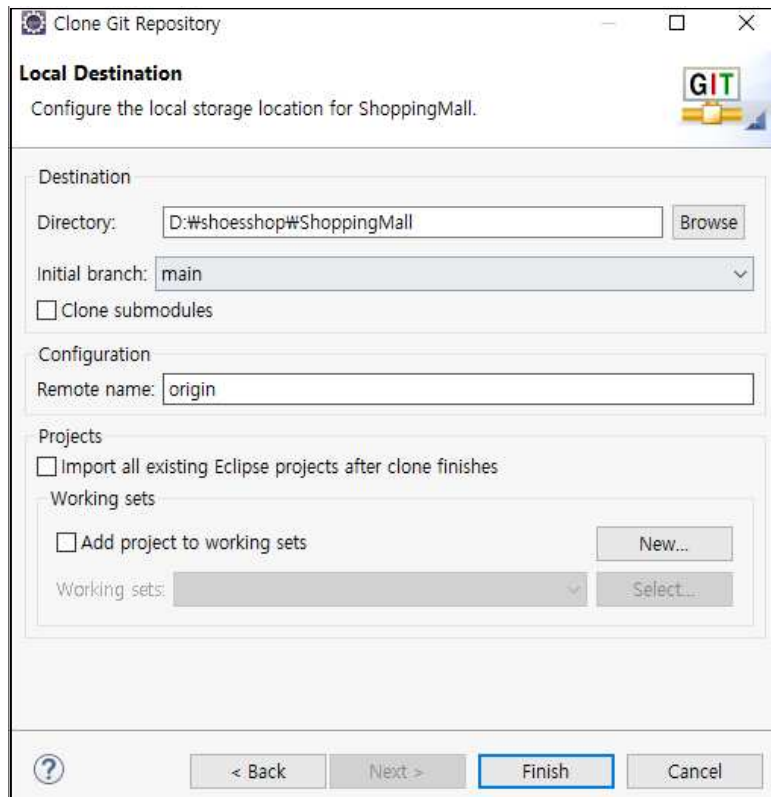
The image shows the 'Clone Git Repository' dialog box. The 'Source Git Repository' section is active. The 'URI' field is highlighted with a red box and contains the text 'v/heejoonk/ShoppingMall.git'. The 'Host' field contains 'github.com' and the 'Repository path' field contains '/heejoonk/ShoppingMall.git'. The 'Connection' section shows 'Protocol' set to 'https'. The 'Authentication' section is highlighted with a red box, showing 'User' as 'heejoonk' and 'Password' as a series of black dots. There is a checkbox for 'Store in Secure Store' which is unchecked. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

- main을 선택하고 Next를 클릭합니다

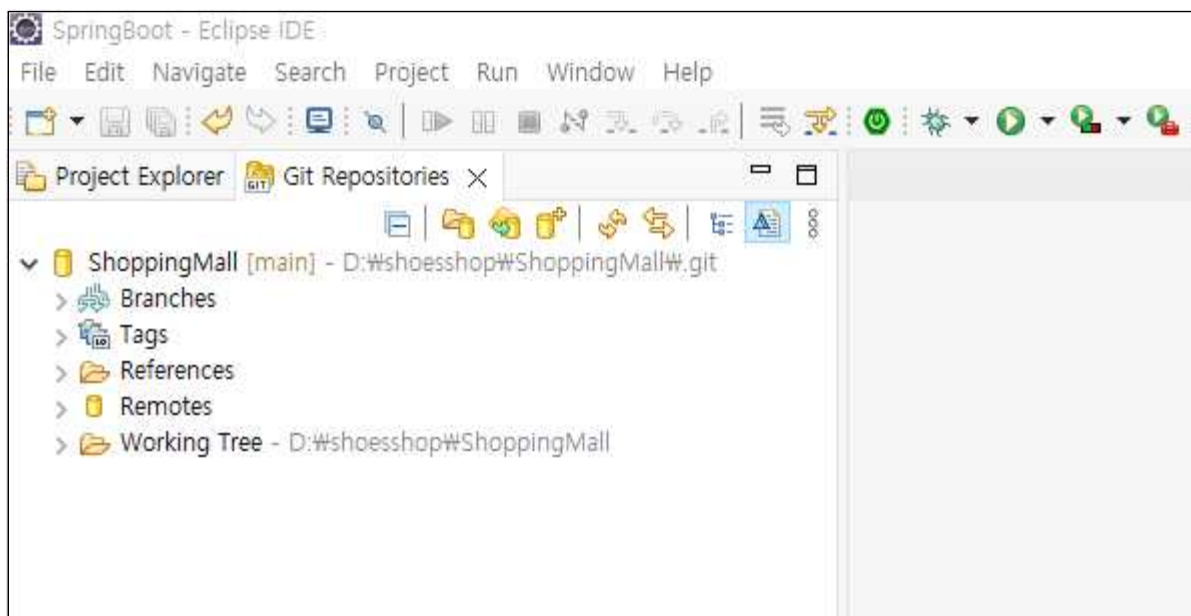


The image shows the 'Branch Selection' dialog box. The title is 'Branch Selection'. The text says 'Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.' Below this, it says 'Branches of https://github.com/heejoonk/ShoppingMall.git:'. There is a 'type filter text' field. Below that, there is a list of branches. The 'main' branch is selected, indicated by a checked checkbox and a red box around it. Below the list, there are buttons for 'Select All' and 'Deselect All'. The 'Tag fetching strategy' section has three radio buttons: 'When fetching a commit, also fetch its tags' (selected), 'Fetch all tags and their commits', and 'Don't fetch any tags'. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a red box.

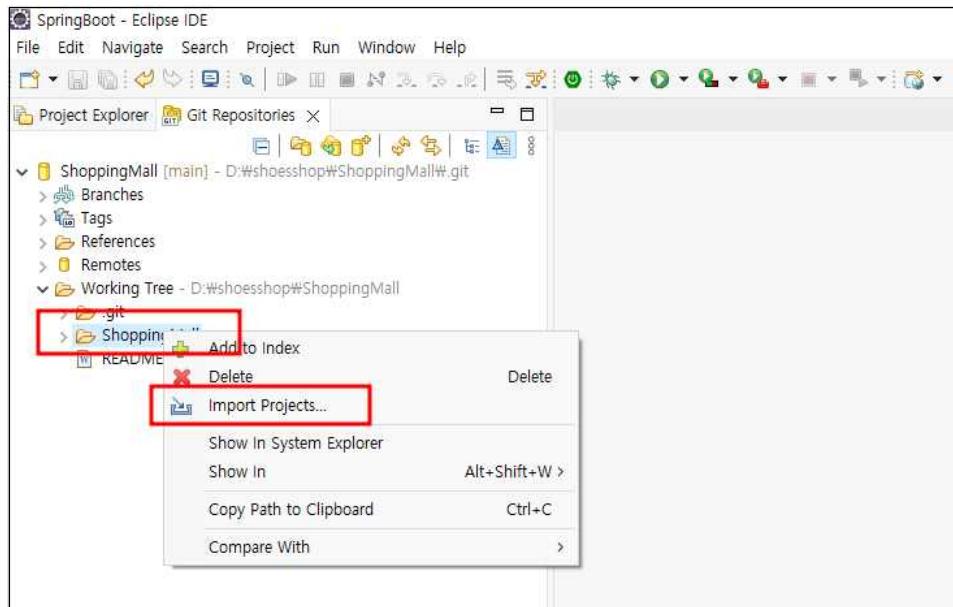
- 다운(PULL)받은 프로젝트를 저장할 Git local 저장장소 (Repository)를 만들고 폴더를 선택합니다. 여기서는 D 드라이브에 shoppingmall 폴더를 만들고 지정하겠습니다.



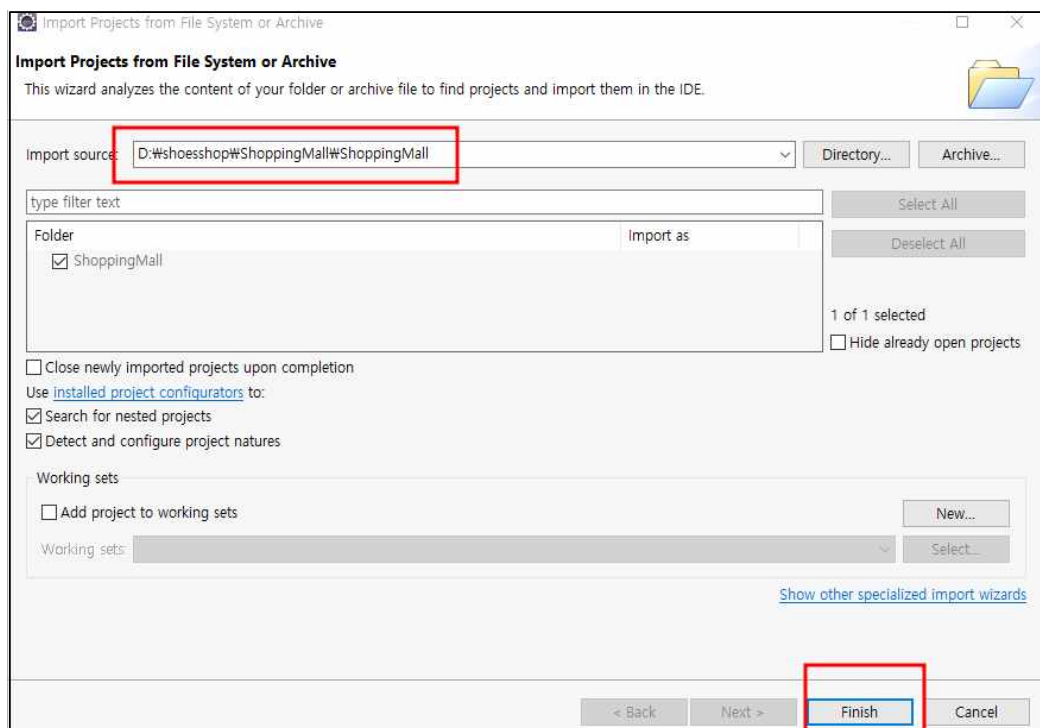
- 그리고 finish를 누르면 local Repository 가 생성됩니다.



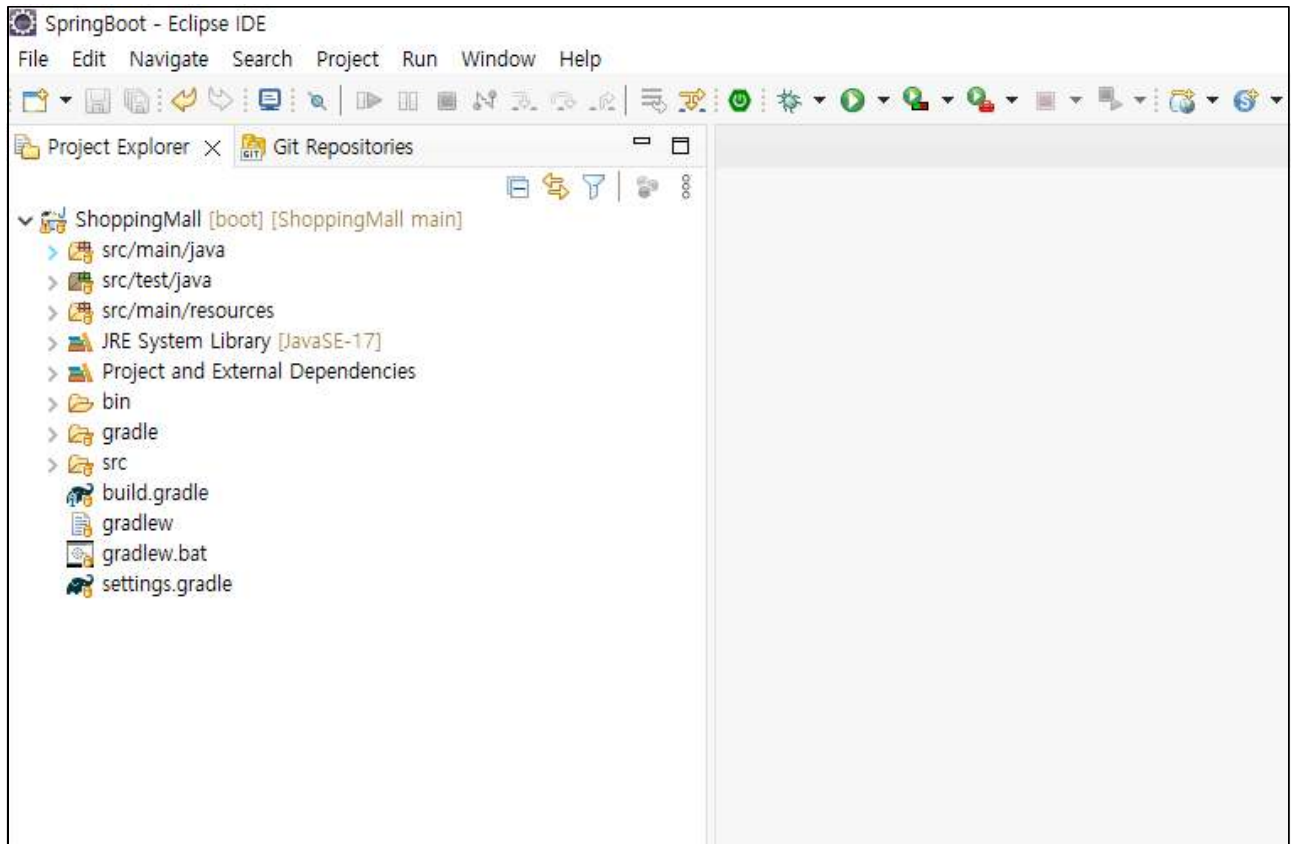
- 이제 이 local Repository에서 현재의 이클립스로 프로젝트를 다운(PULL)받아 옵니다.
- local repository를 펼치서 나오는 Working Tree를 펼치고 프로젝트 이름을 확인 후 프로젝트 일므위에서 오른쪽 버튼누르고 Import project를 선택합니다



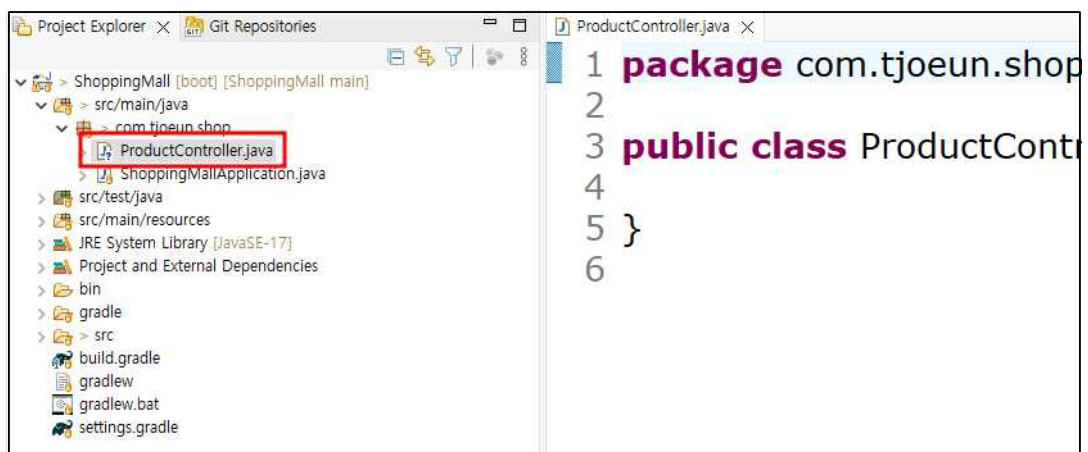
- 프로젝트 import 내역을 확인하고 finish를 누르면 작업할 project 가 project explorer 에 импорт 됩니다



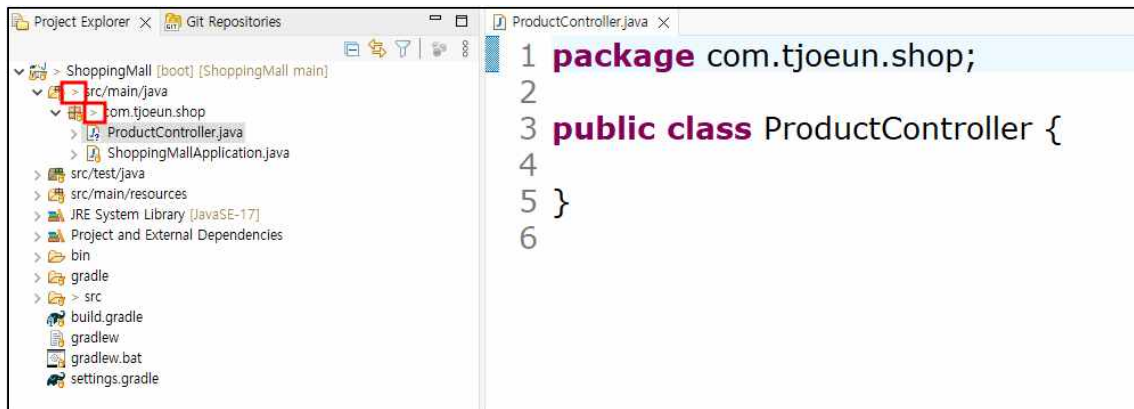
■ 최종 импорт 화면



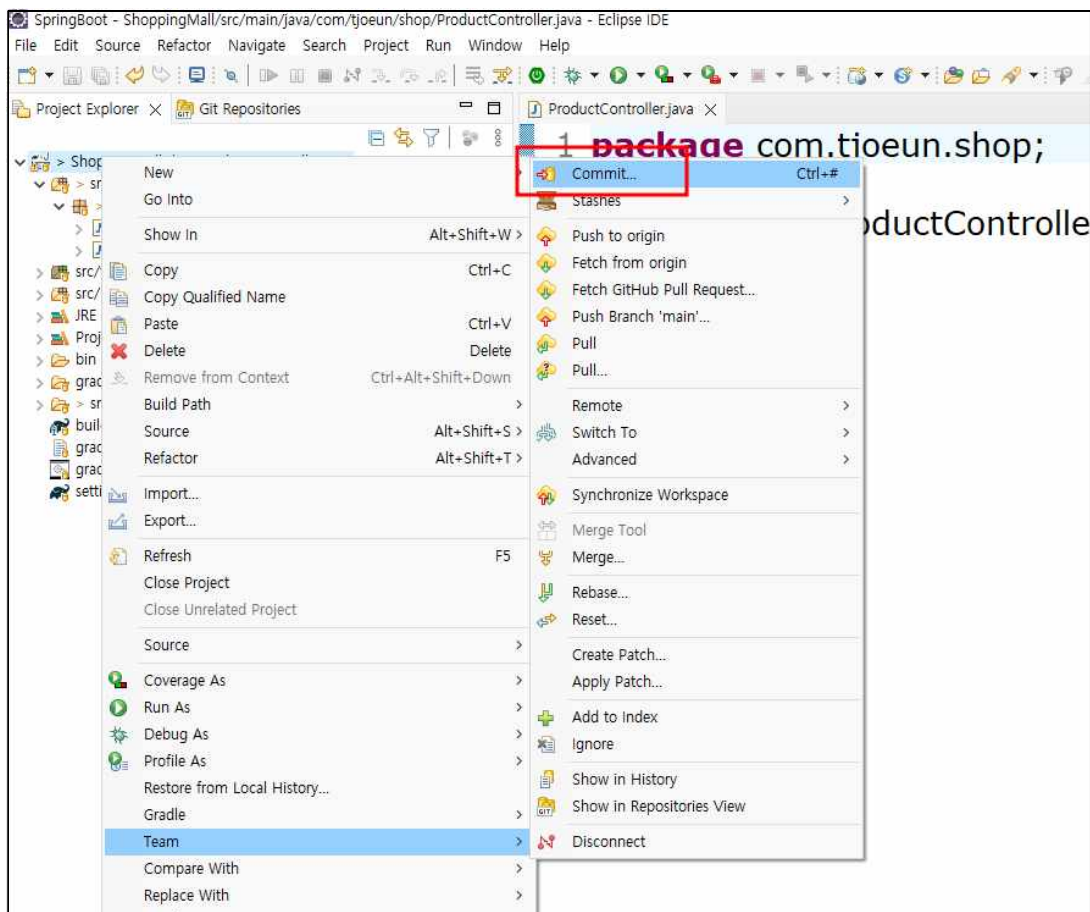
- 이제 프로젝트 이름위에서 오른쪽 버튼을 누르고 Team 메뉴에서 내가 변경한 내용을 GitHub 에 PUSH 할 수 있습니다. 먼저 src/main/java 안의 기본 패키지인 com.tjoeun.shop에 ProductController.java 파일을 만들겠습니다



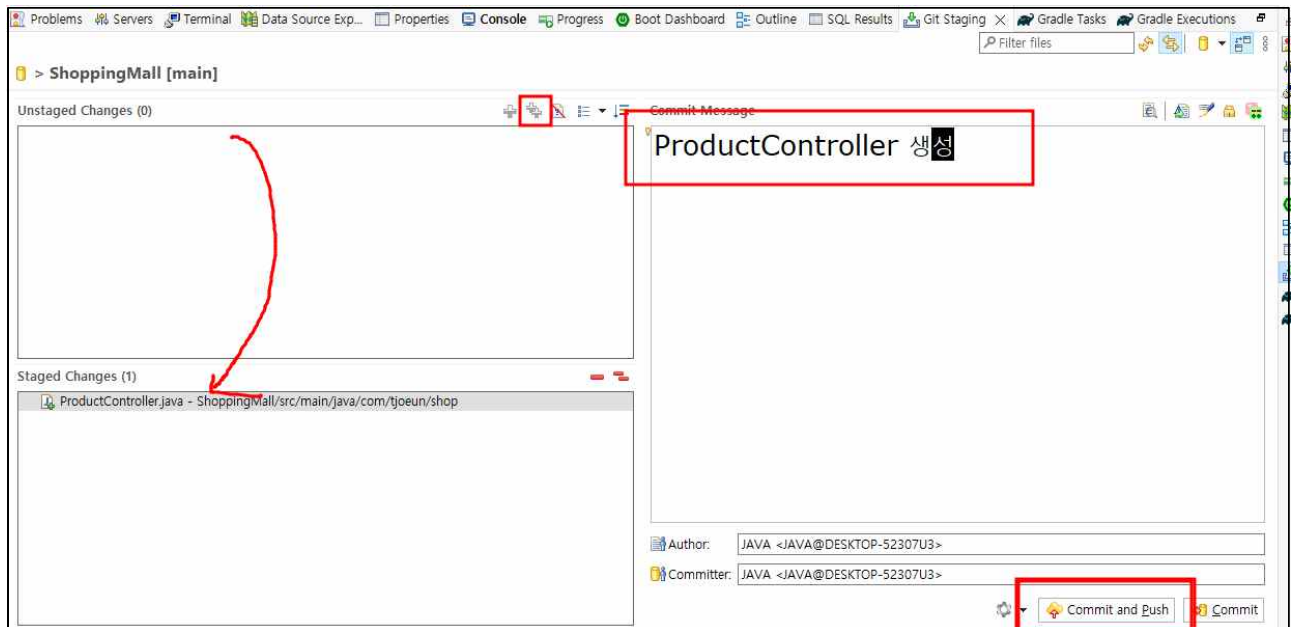
- 변경이 감지된 이클립스는 변경된 파일과 그를 포함하는 폴더에 변경이 있음을 표시합니다



- 이들은 프로젝트 이름위에서 오른쪽 버튼을 누르고 Team - Commit 메뉴를 이용하여 GitHub 로 전송합니다



- 이때 Staging 화면에서 위쪽 Unstaged Changes 안의 파일 (들)을 Staged Changes 로 옮기고, 반드시 Commit Message 를 입력해줍니다



- commit and push를 눌렀을 때 나오는 로그인창에는 아이디와 토큰키를 입력합니다. 그리고 아래와 같은 화면이 표시되면 정상적으로 push가 완료된 상태입니다

