

깃허브 프로젝트 관리

깃허브 프로젝트 관리

• 참고

PR이 승인되어 Merge된 내용이 있다면 팀원들에게 반드시 공유 해줘야 함
충돌이 한번에 여러 개 발생하면 해결하기 어렵기 때문

PR이 승인되어 Master 브랜치에 변동 사항이 생겼을 경우

1. 작업중이던 내용 백업
→ 팀원들은 작업하던 내용 저장 후 자신의 브랜치에 Commit And Push
2. Master 브랜치의 코드를 가져와 업데이트
→ Master 브랜치를 Pull 받아온 후 충돌되는 내용이 있다면 해당 코드를 작성한 팀원과 상의하여 충돌 해결
3. 자신의 브랜치에 Master 브랜치의 변경사항 병합
→ 문제가 더 이상 없다면 자신의 브랜치에 Commit And Push 후 다시 Pull 받아옴
→ Master 브랜치와 내 브랜치의 내용이 합쳐짐

깃허브 프로젝트 관리 (제약 설정)

제약 설정 - 1

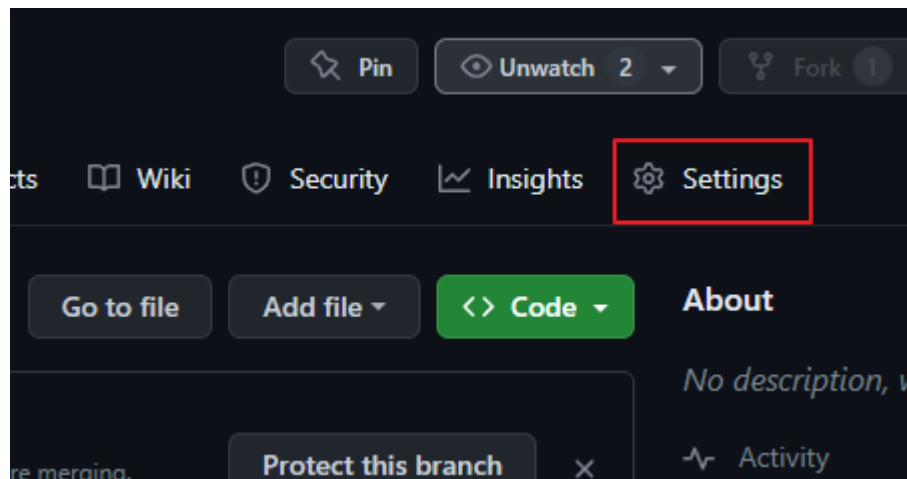
- 제약 설정

아래에서 설정할 제약 설정은 PR을 요청했을 때 몇 명이 승인해야 merge를 진행 시킬지에 대한 룰을 만드는 것

제약 설정 - 2

- 설명

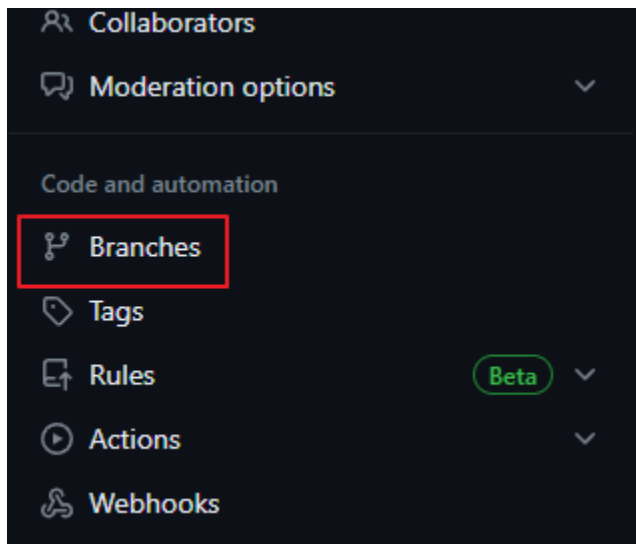
깃허브의 프로젝트 들어가서 Settings 클릭



제약 설정 - 3

- 설명

우측 Branches 클릭



제약 설정 - 4

- 설명

Add branch protection rule 클릭

Branch protection rules



You haven't protected any of your branches

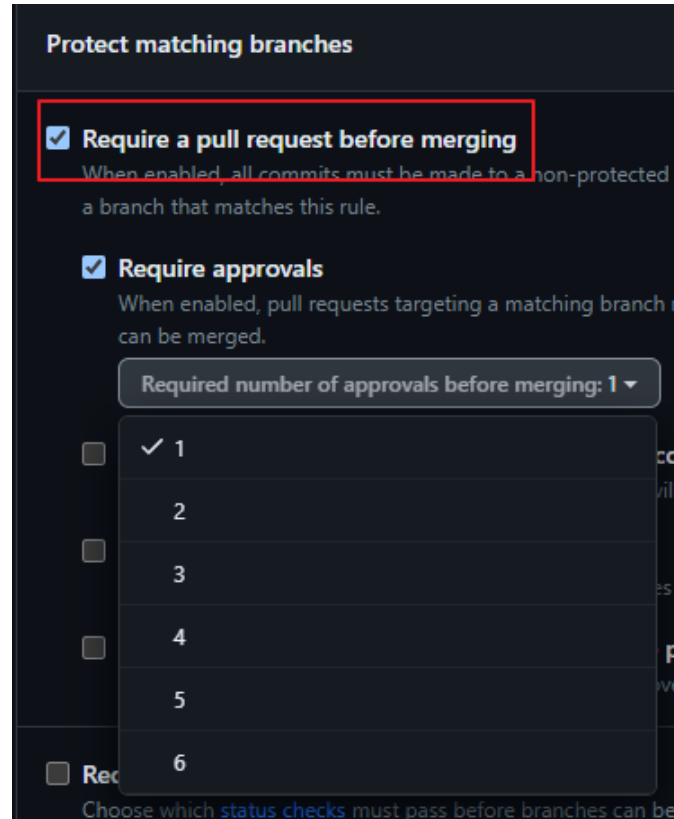
Define a protected branch rule to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. [Learn more about protected branches](#)

Add branch protection rule

제약 설정 - 5

- 설명

몇 명의 팀원에게 PR 승인을 받아야 master 브랜치에 합칠건지 선택 후 드래그 맨 밑으로 내려서 Create

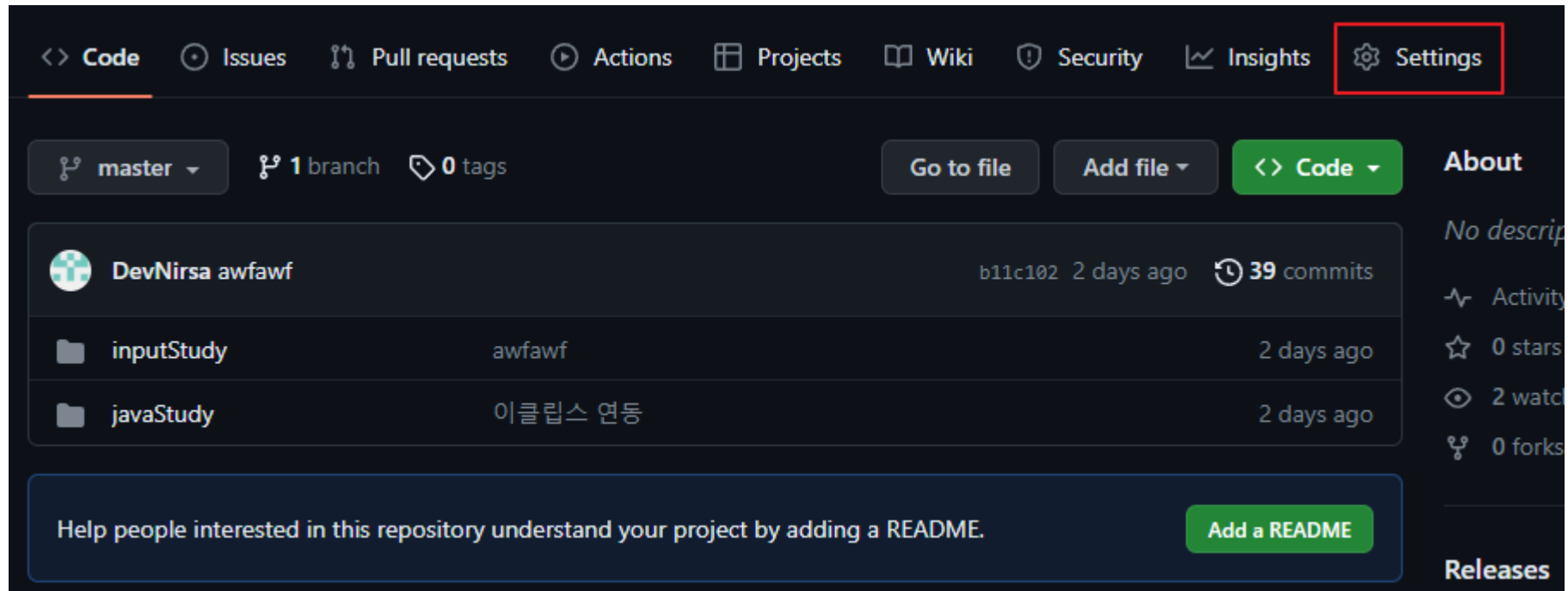


Collaborator 초대
(팀장 또는 깃허브 관리자)

Collaborator 초대 - 1

- 프로젝트 관리자

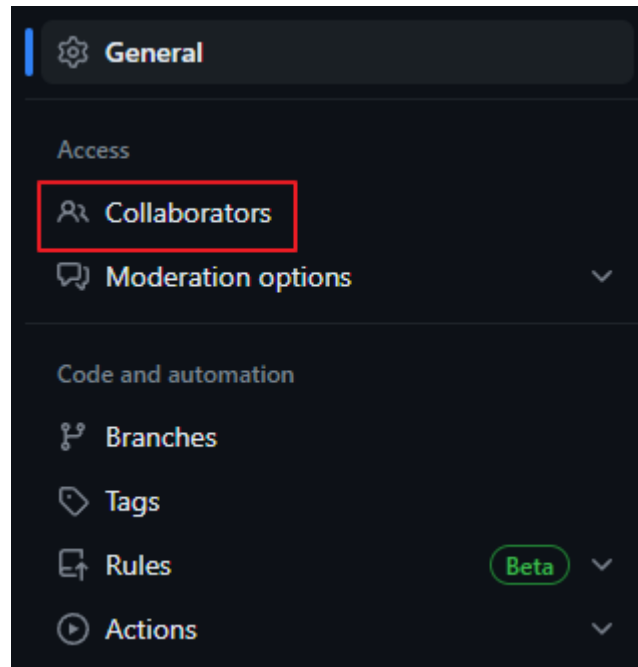
깃허브 내 프로젝트까지 접속 후 Settings 클릭



Collaborator 초대 - 2

- 프로젝트 관리자

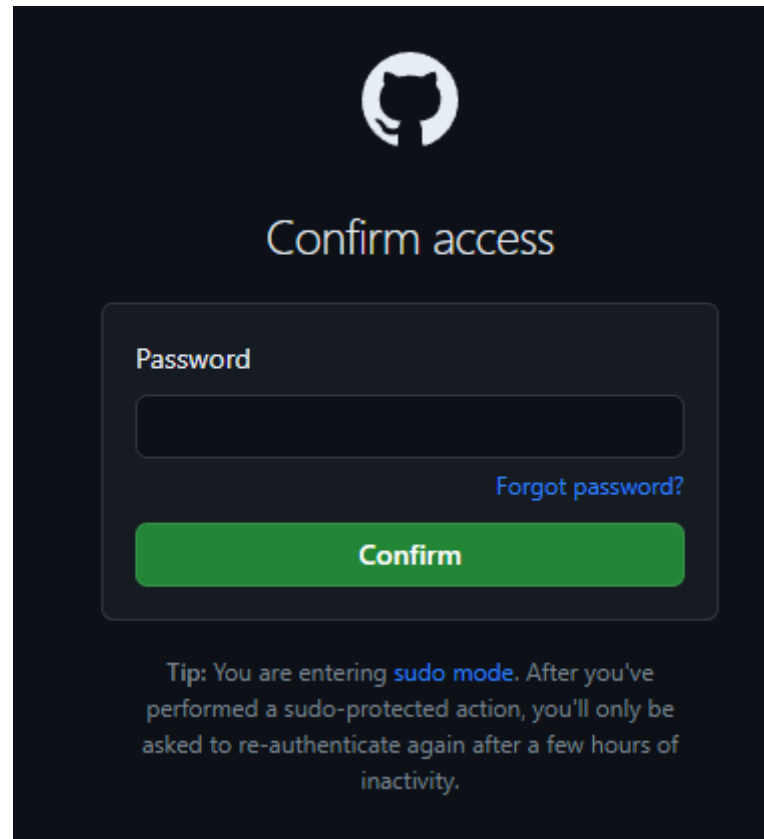
우측 메뉴에 Collaborators 클릭



Collaborator 초대 - 3

- 프로젝트 관리자

만약 패스워드를 입력하라는 창이 나오면 깃허브 패스워드 입력

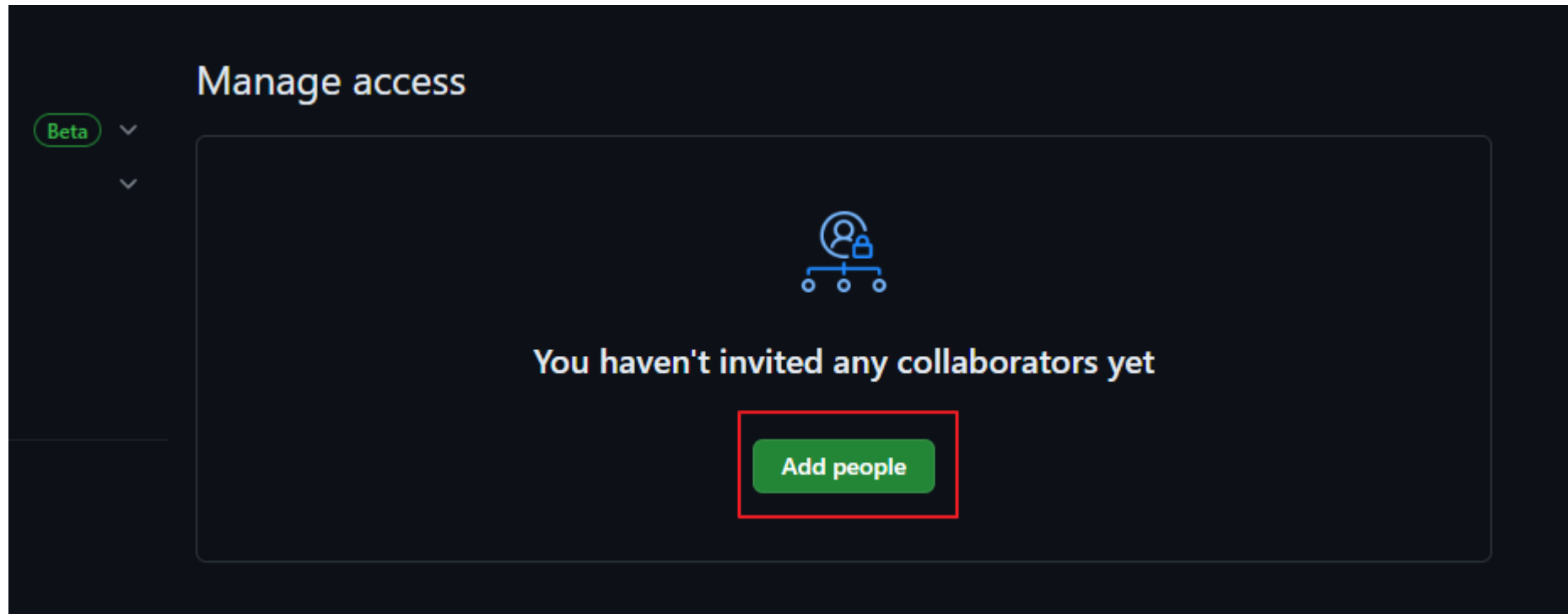


The image shows a dark-themed GitHub interface for confirming access. At the top is the GitHub Octocat logo. Below it, the text "Confirm access" is centered. A form box contains a "Password" label, a text input field, and a "Forgot password?" link. A green "Confirm" button is at the bottom of the form. Below the form, a tip message states: "Tip: You are entering `sudo mode`. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity."

Collaborator 초대 - 4

- 프로젝트 관리자

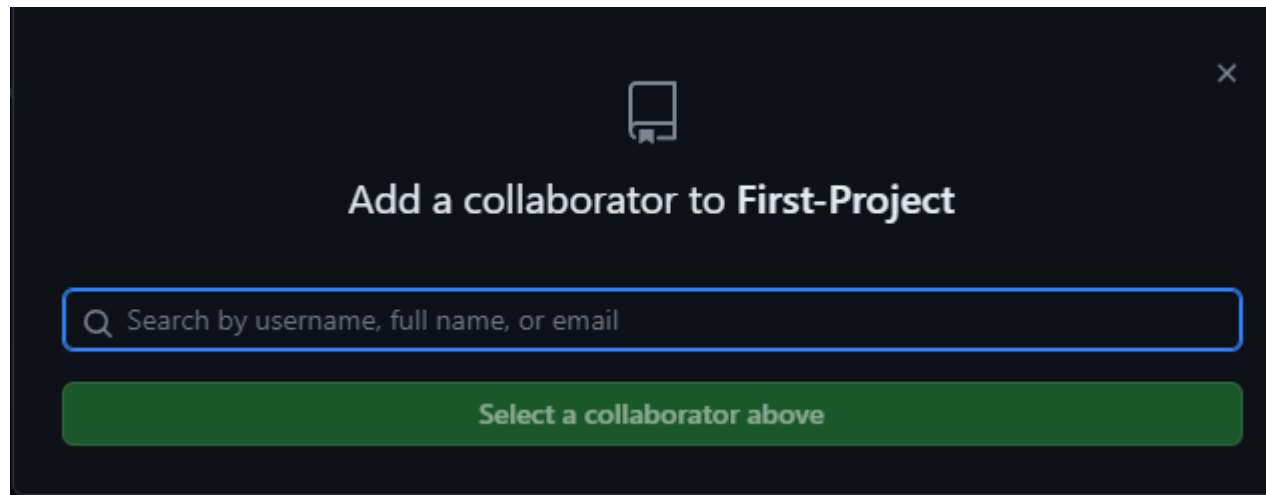
Add people 클릭



Collaborator 초대 - 5

- 프로젝트 관리자

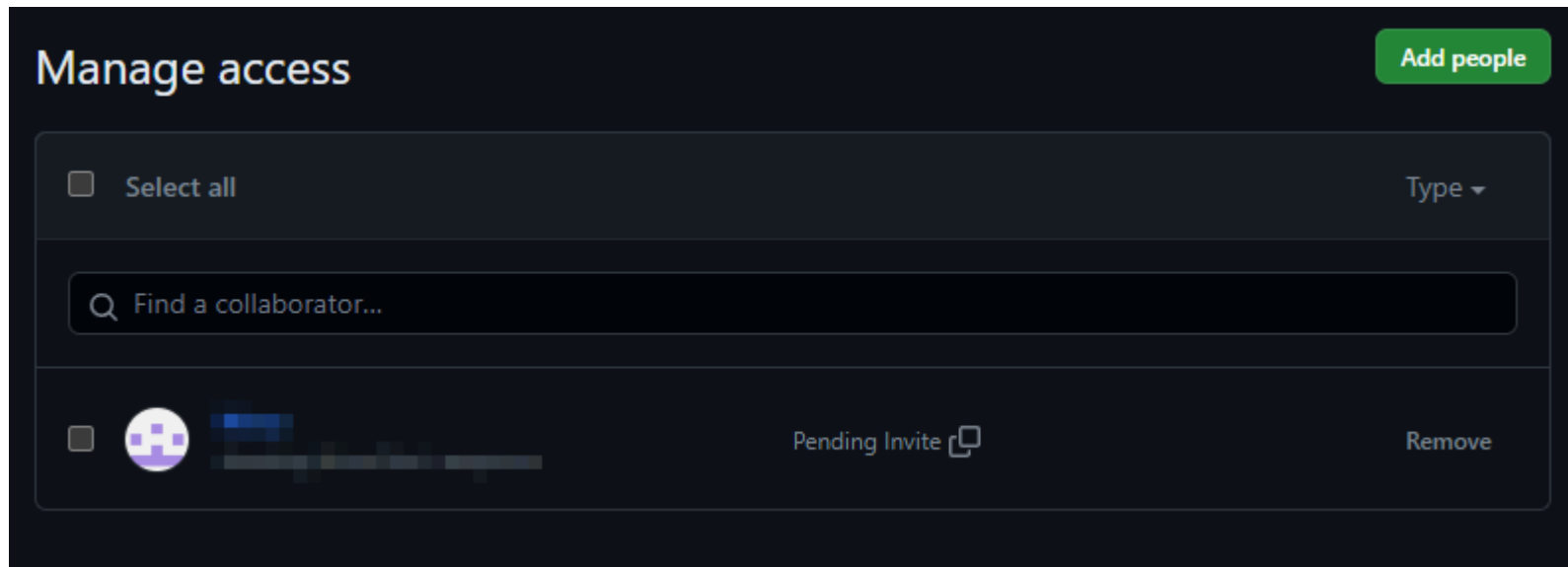
팀원 이메일 입력 후 초대



Collaborator 초대 - 6

- 프로젝트 관리자

아래와 같이 추가된게 보이면 성공

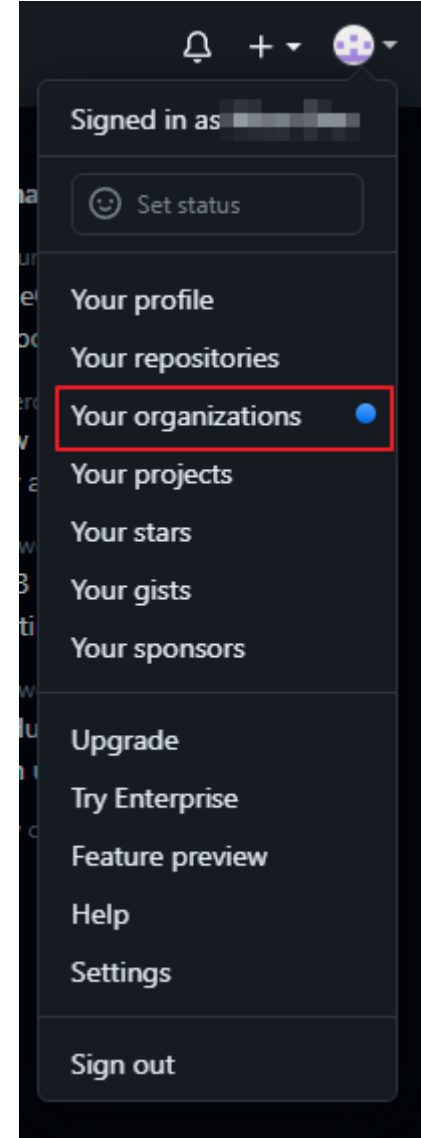


Collaborator 수락하기 (팀원 또는 이용자)

Collaborator 수락하기 - 1

- 팀원

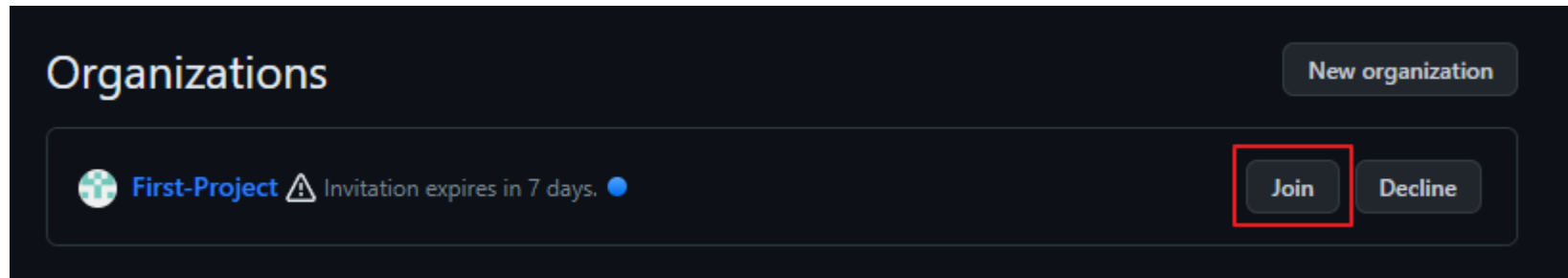
1. 깃허브 홈페이지에서 우측 상단 프로필 클릭
2. Your organizations 클릭



Collaborator 수락하기 - 2

- 팀원

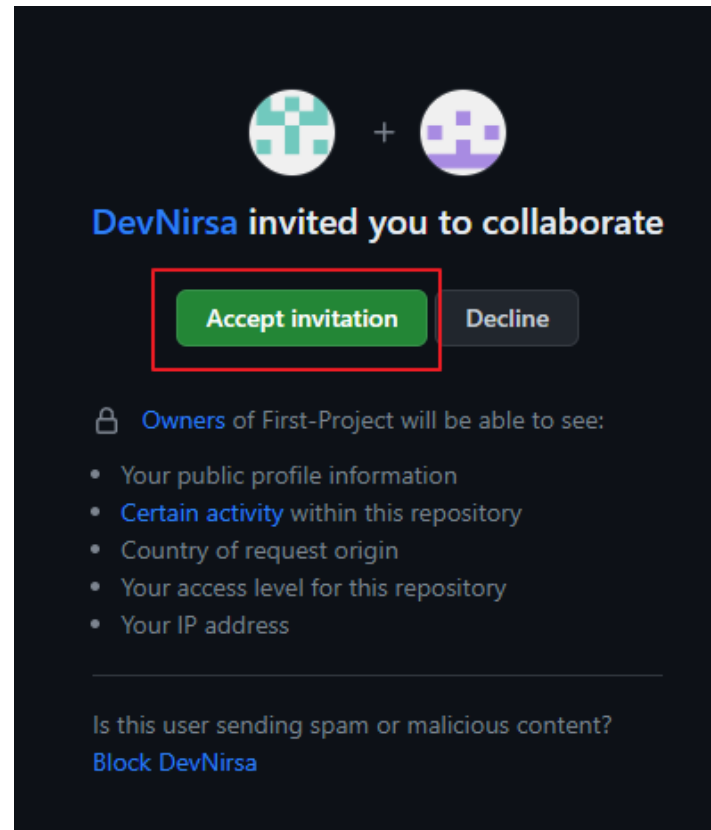
Join 클릭



Collaborator 수락하기 - 3

- 팀원

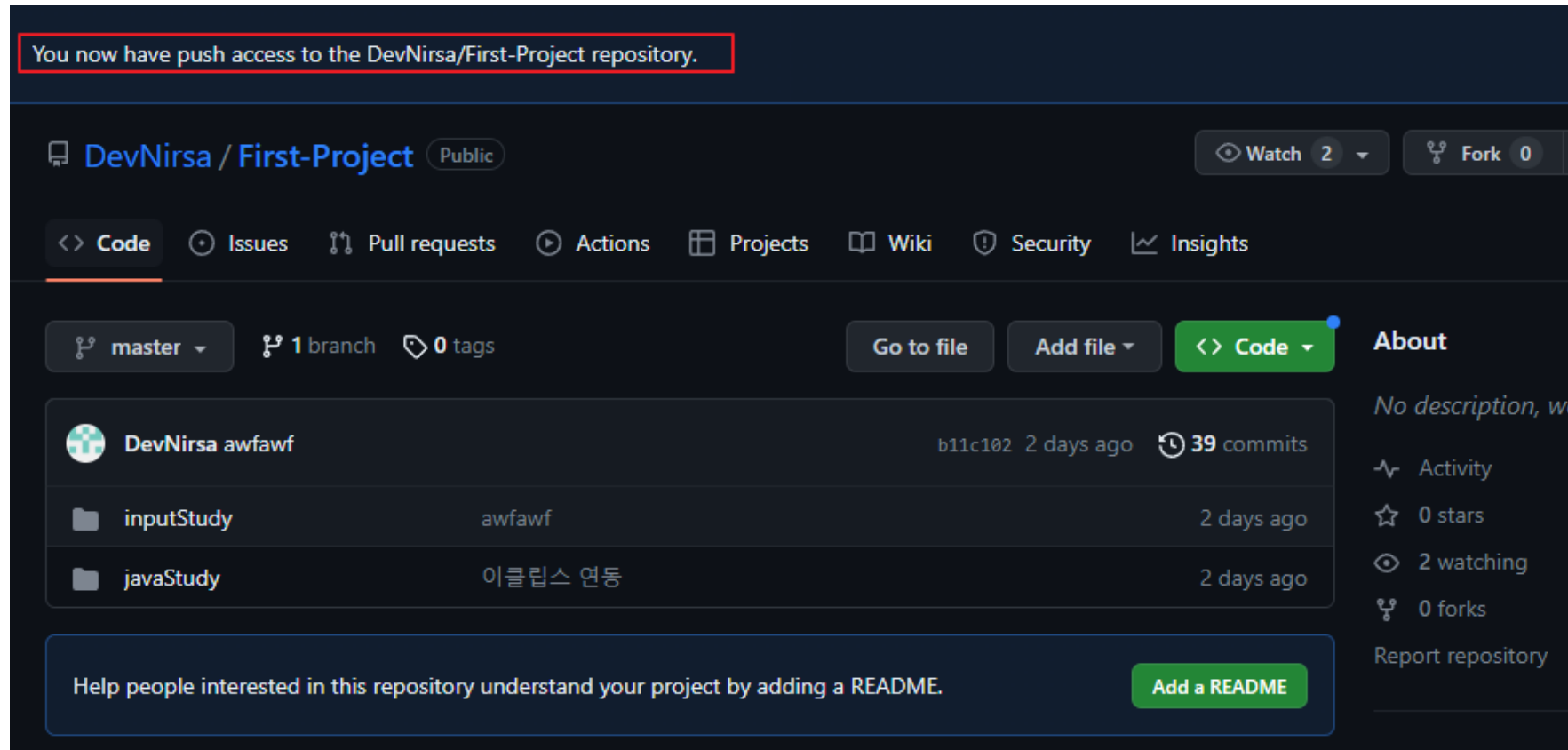
Accept invitation 클릭



Collaborator 수락하기 - 4

- 팀원

아래와 같은 메시지가 나오면 성공

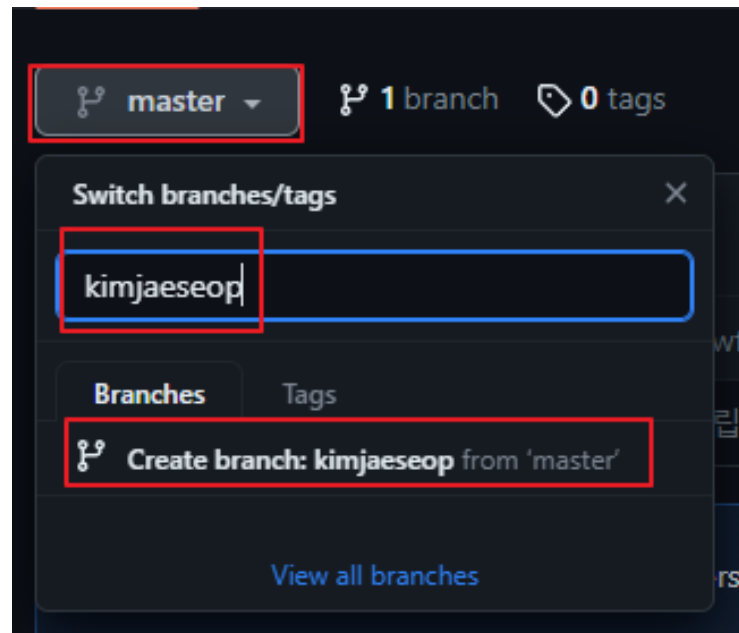


PR 올려보기 (팀원, VSCode)

PR 올려보기 – VSCode - 1

- 팀원

권한을 받은 프로젝트에 내가 작업할 브랜치 생성

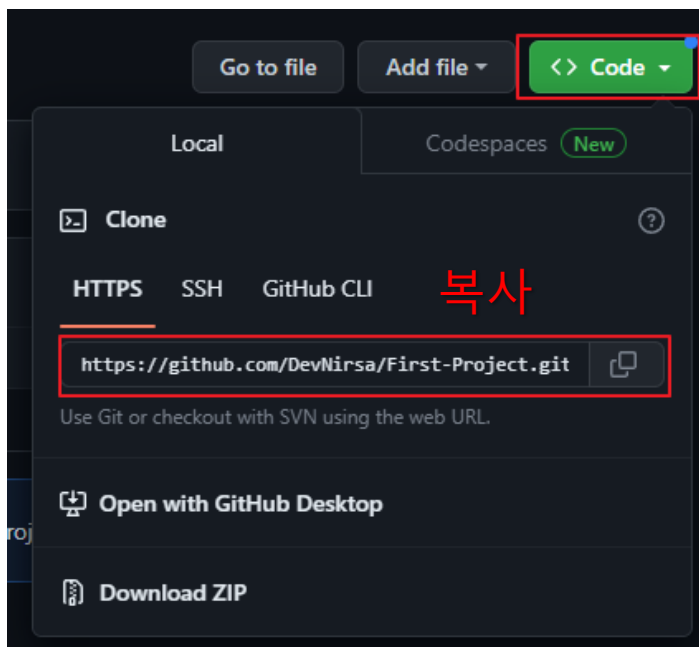


PR 올려보기 - VSCode - 2

- 팀원

아래와 같이 명령어 입력

* 작업할 디렉토리 생성 후 폴더 열기



```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)
```

```
$ git init
```

```
Initialized empty Git repository in C:/Test/.git/
```

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)
```

```
$ git remote add origin https://github.com/DevNirsa/First-Project.git
```

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)
```

```
$ git pull origin master
```

```
remote: Enumerating objects: 171, done.
```

```
remote: Counting objects: 100% (171/171), done.
```

```
remote: Compressing objects: 100% (110/110), done.
```

```
Receiving objects: 50% (86/171)remote: Total 171 (delta 42), reused 144 (delta 26), pack-reused 0
```

```
Receiving objects: 100% (171/171), 26.50 KiB | 6.63 MiB/s, done.
```

```
Resolving deltas: 100% (42/42), done.
```

```
From https://github.com/DevNirsa/First-Project
```

```
* branch          master    -> FETCH_HEAD
```

```
* [new branch]     master    -> origin/master
```

PR 올려보기 - VSCode - 3

- 팀원

브랜치 생성 후 이동

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)  
$ git branch kimjaeseop
```

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)  
$ git switch kimjaeseop  
Switched to branch 'kimjaeseop'
```


PR 올려보기 - VSCode - 4

- 팀원

브랜치 생성 후 이동

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)  
$ git branch kimjaeseop
```

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (master)  
$ git switch kimjaeseop  
Switched to branch 'kimjaeseop'
```

PR 올려보기 - VSCode - 5

- 팀원

파일 수정 후 깃허브의 kimjaeseop 브랜치에 push

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (kimjaeseop)  
$ git add .
```

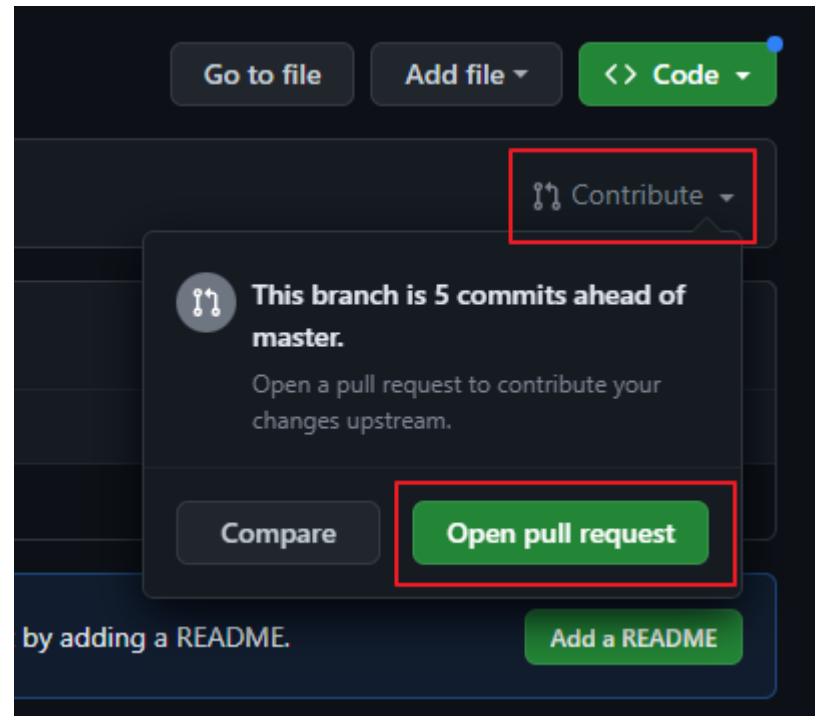
```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (kimjaeseop)  
$ git commit -m "PR"  
[kimjaeseop 0c15484] PR  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
GR803@DESKTOP-SQL4H5D MINGW64 /c/Test (kimjaeseop)  
$ git push origin kimjaeseop  
Enumerating objects: 48, done.  
Counting objects: 100% (48/48), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (35/35), done.  
Writing objects: 100% (40/40), 2.81 KiB | 958.00 KiB/s, done.  
Total 40 (delta 20), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (20/20), completed with 5 local objects.  
To https://github.com/DevNirsa/First-Project.git  
b11c102..0c15484 kimjaeseop -> kimjaeseop
```

PR 올려보기 – VSCode - 6

- 팀원

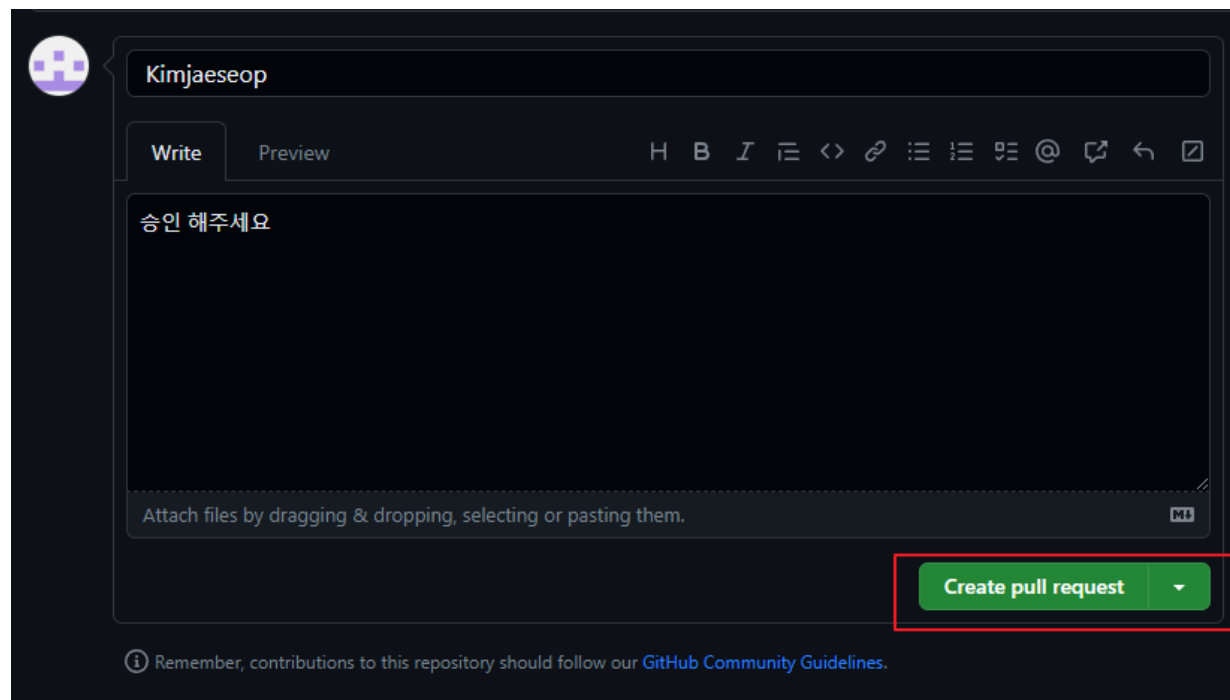
깃허브 프로젝트에 만든 내 브랜치에서 Contribute → Open Pull Request



PR 올려보기 – VSCode - 7

- 팀원

PR 요청

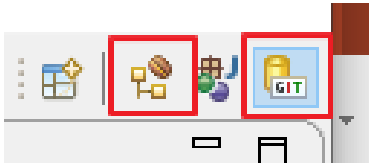


PR 올려보기 (팀원, Eclipse)

PR 올려보기 – Eclipse - 1

- 참고

자바 프로젝트가 있는 곳과 깃은 우측 상단의 아이콘을 통해 이동





PR 올려보기 – Eclipse - 2

- 팀원

권한을 받은 프로젝트에 내가 작업할 브랜치 생성

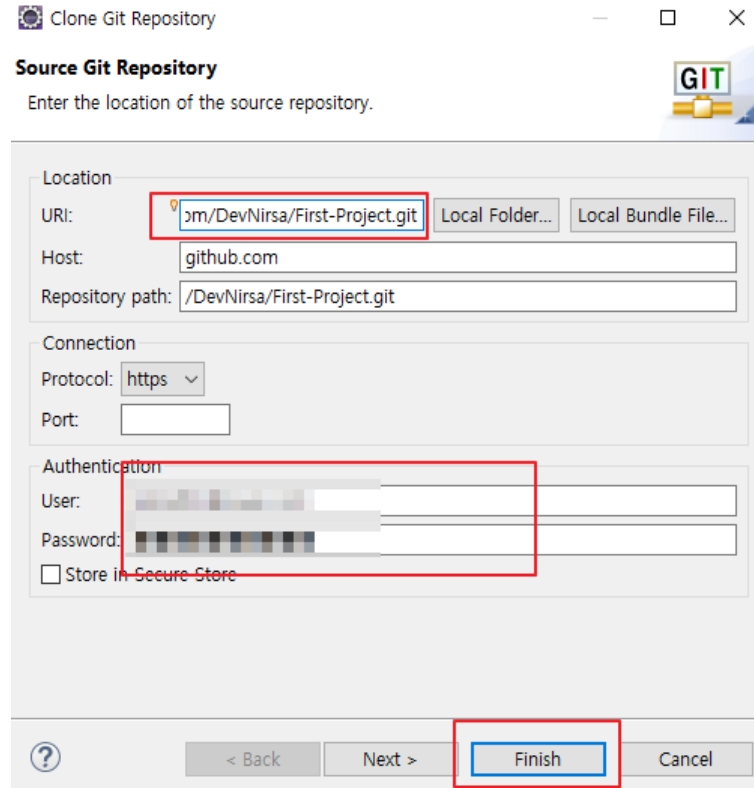
Select one of the following to add a repository to this view:

-  [Add an existing local Git repository](#)
-  [Clone a Git repository](#)
-  [Create a new local Git repository](#)

PR 올려보기 – Eclipse - 3

- 팀원

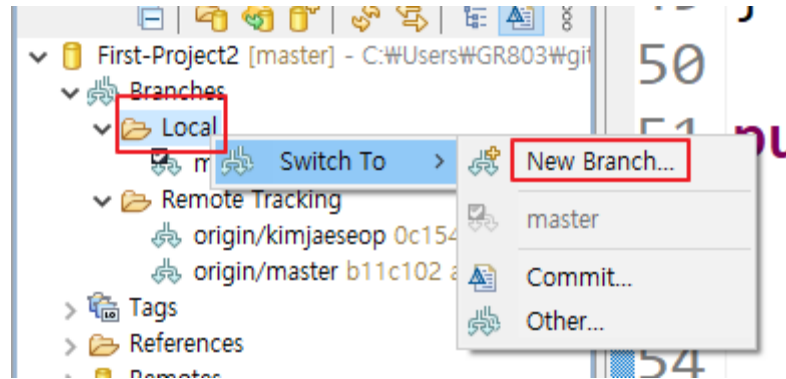
URL과 계정 정보 입력 후 피니쉬



PR 올려보기 – Eclipse - 4

- 팀원

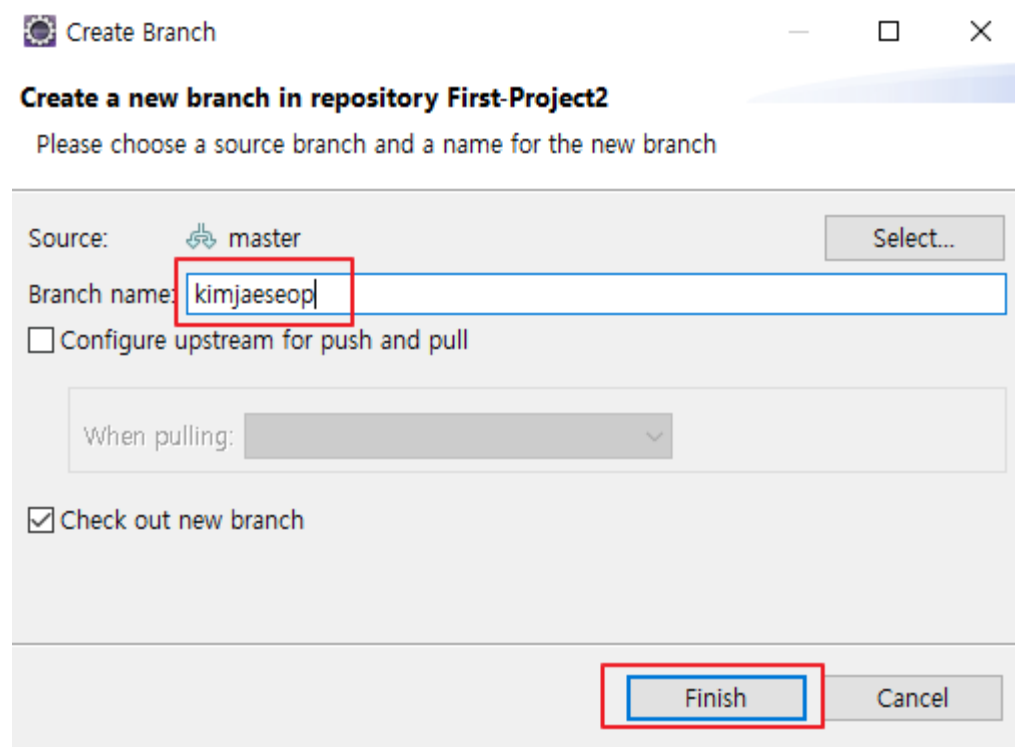
Local 우클릭 → Switch To → New Branch...



PR 올려보기 – Eclipse - 5

- 팀원

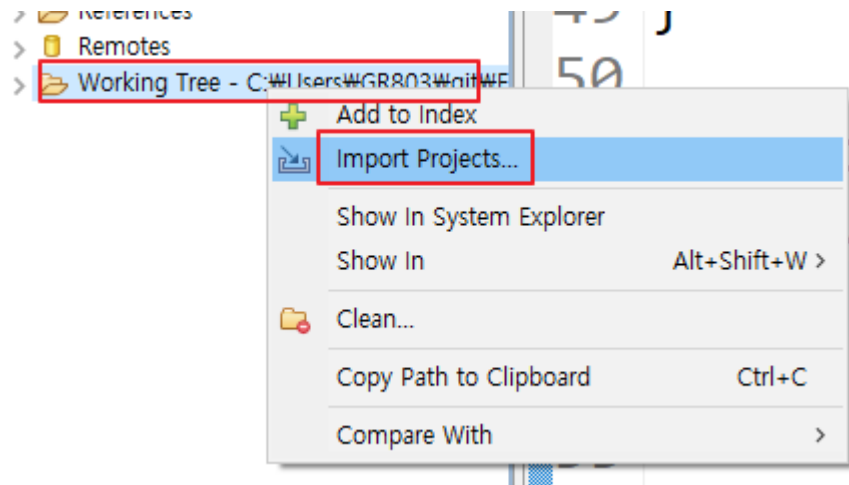
브랜치 이름 입력 후 Finish



PR 올려보기 – Eclipse - 6

- 팀원

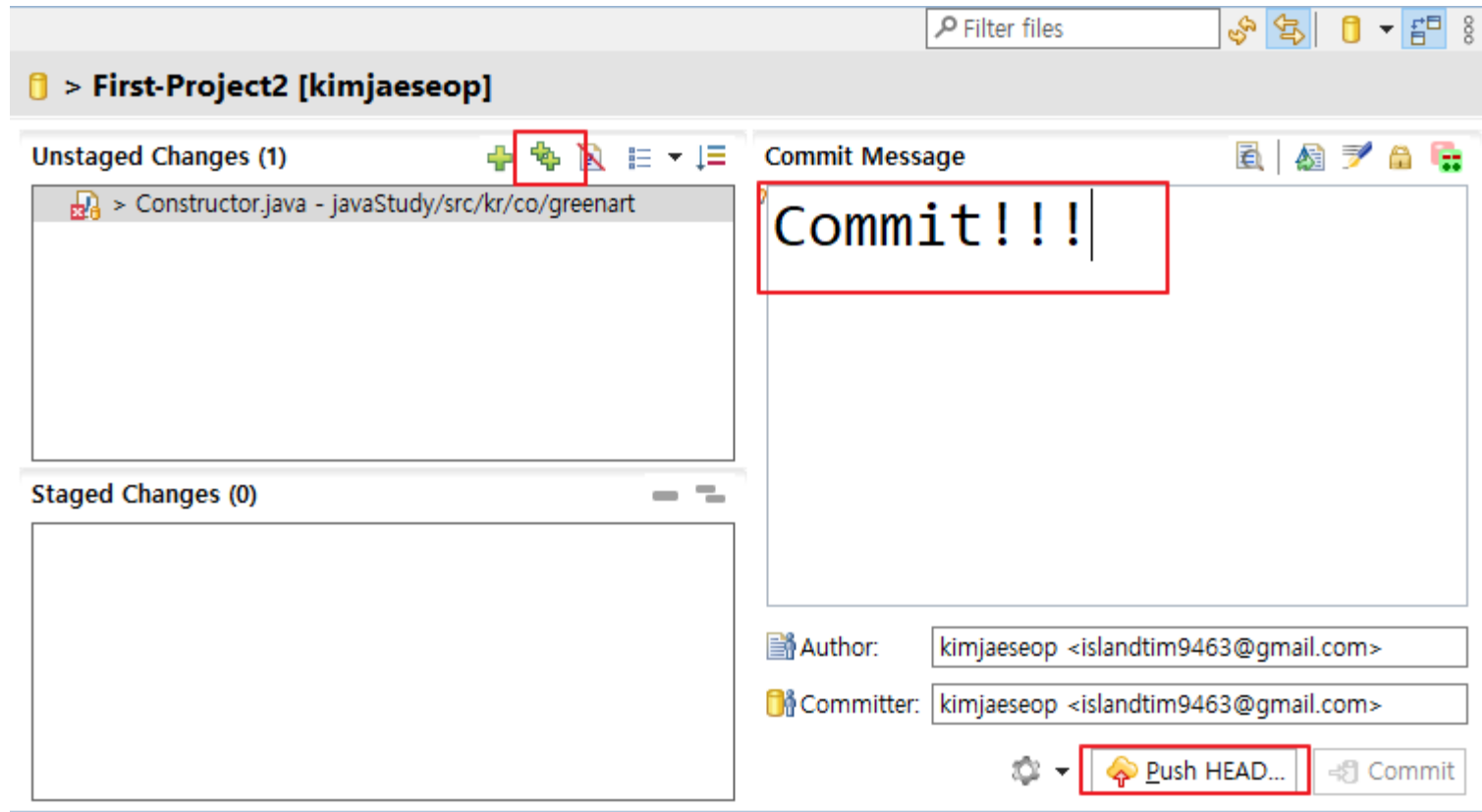
Working Tree → Import Projects... → Finish



PR 올려보기 – Eclipse - 7

- 팀원

Java Project Expolorer로 돌아와서 생긴 프로젝트의 내용을 수정 후 Commit, Push 테스트

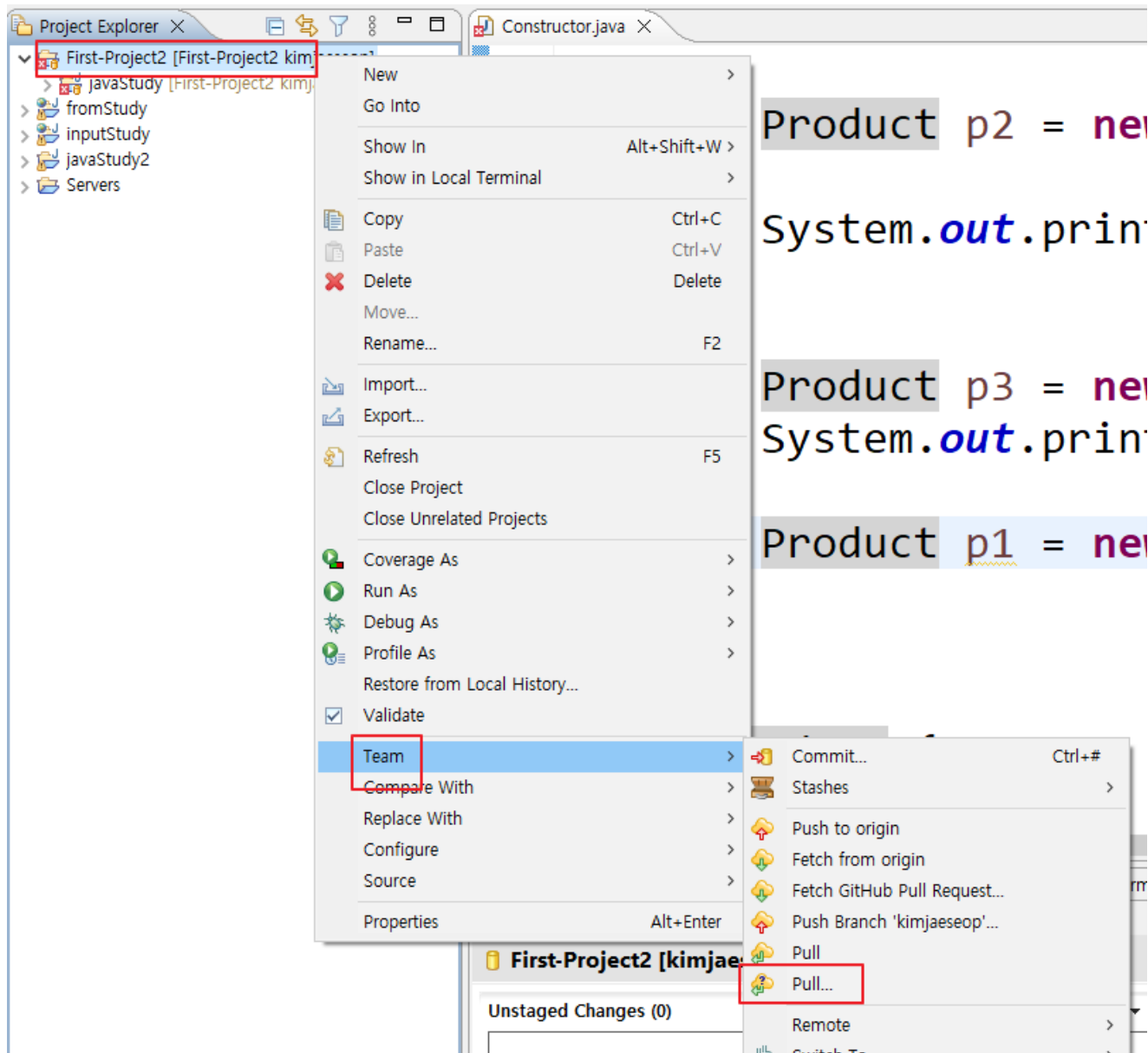


Master 브랜치 Pull 받기 (Eclipse)

Master 브랜치 Pull 받기

• 설명

1. 프로젝트 우클릭
2. Team
3. Pull...



Master 브랜치 Pull 받기

- 설명

Refence에 master 입력하면 밑에 master [branch] 가 나오는걸 클릭 후 Finish

