



gerrit 사용법

▶ 게릿 번호 (ssafy0x)

| 참고: project SSIFY - Help - 매뉴얼 - Gerrit 매뉴얼

- 🔴 게릿에 깃랩 내용들이 다 알아서 복사돼있는 느낌
- 게릿에 푸시하고 하면 알아서 게릿이 깃랩에 넣어주는 느낌
- 게릿장점: 커밋단위 리뷰
- 파트너가 코드리뷰를 하고 난 뒤에 승인을 해주어야 변경사항이 깃랩에 반영되는 구조
- 깃랩에 직접 push를 하면 에러가 남
- 빈브랜치 올리고 - 자기거 올리고 -머지되면삭제하고 (+푸시할땐 리뷰하고)

Gerrit URL : `https://k10e206.p.ssafy.io:8989/`
ID : admin
pw : aksmfQkd6!

계정 선택 (ssafy01내꺼임) ->
settings ->
full name, display name 변경

email addresses 이메일 입력후 send verification -> 따로 메일 인증없이 적용됨

generate new password 버튼 클릭 -> 발급된 토큰 저장해놓기

그냥 바로 클론할 때
`git clone "https://{ 내 게릿 유저 네임 ex) ssafy00 }@k10e206.p.ssafy.io:8989/a/S10P22E206"`

게릿 연결
`git remote set-url origin "https://ssafy04(여기에 자기에 맞는 번호)@k10e206.p.ssafy.io:8989/`

review 오픈할 때
`git push <remote명> HEAD:refs/for/<branch명>`

주의사항:

1. gitlab에 직접 푸쉬 하지말기
 2. 리뷰 오픈하기전에 브랜치가 생성되어 있어야함(브랜치 생성하자마자 바로 푸쉬해놓기)
- `git checkout -b 브랜치명 || git switch -c 브랜치명`
`git push origin 브랜치명`

1. 이메일 등록하기

Email Addresses

Email

Preferred

DELETE

SAVE CHANGES

New email address

email@example.com

SEND VERIFICATION

2. BROWSE - Repositories - RepositoryName 클릭
3. 혹은 걸 수 있는 clone with commit-msg hook 명령어 복사
 - a. email 인증 필요
 - b. HTTP Credentails - generate new pwd
 - c. 발급해주는거 사용하면 됨
4. git bash 열고 clone 하기
5. 브랜치 만들어서 작업을 하려면 git checkout -b BE-XX/작업내용으로 브랜치를 만들기
 - a. 브랜치 만들기 전 기반이 될 브랜치로 이동 `git switch BE-dev`
 - b. 로컬에서 브랜치 만들기 `git checkout -b BE-JW/작업내용`
 - c. 이걸 원격에 push를 해주어야 함 `git push origin HEAD`
6. 변경사항 작업 후 add → commit → push" 이후 `git push origin HEAD:refs/for/BE-XX/작업내용` 하면 리뷰가 올라감
 - a. 그냥 git push origin HEAD 하면 리뷰가 안 들어감
7. 웹에서 리뷰 승인 후
 - a. CHANGES - Open - Reviewers를 눌러서 리뷰하기
 - b. -2점을 주면 누가 +2를 해도 머지할 수 없음
 - c. reject 당할 경우 Change-Id를 이용해서 수정 후 다시 커밋할 수 있음 (진우오빠는 안해봄)
8. BE-dev 브랜치로 다시 가서 merge
 - a. 머지 할 브랜치로 이동
 - b. 머지하기 `git merge BE-XX/작업내용`
 - c. 충돌은 여기서 잡는다!

게리트를 잘 사용해보자!!

예를 들어 수빈이가 FE-develop 에서 branch 를 FE-SB/login 을 파놓기만 했다 !!

```
cd FE-develop
git checkout -b FE-SB/login
```

1. 바로 push 를 한다 (게릿에는 수빈이가 판 FE-SB/login 브랜치가 없기 때문 !)

```
git push origin FE-SB/login{브랜치명}
```

2. 그리고 신나게 기능개발을 한다.
3. add, commit 을 한다.

```
git add *  
git commit -m "Feat : ~~ "
```

4. push 를 한다.

- a. 코드리뷰 받고 싶다.

```
git push origin HEAD:refs/for/FE-SB/login{<-- 이 자리에 브랜치명}
```

- b. 코드리뷰 받고 싶지 않아요

```
git push origin FE-SB/login{<-- 이 자리에 브랜치명}
```

코드리뷰를 다 받고 난 후에 !! submit 버튼까지 누르고 !!! 아래의 머지를 시작함

5. 이제 merge를 한다.

- merge 를 2번을 해야합니다. FE-develop 1번 + FE 에 한번

```
git checkout FE-develop  
git pull  
git merge FE-SB/login{<-- 이 자리에 브랜치명}  
git push origin FE-develop  
  
git checkout FE  
git merge FE-develop  
git push origin FE git
```

이렇게까지 하고 게릿에 확인

6. 브랜치 삭제

```
git push origin -d FE-SB/login
```

로컬 브랜치 삭제

```
git branch -d FE-SB/login
```

원격이랑 bash랑 git branch 동기화

```
git fetch --all --prune
```