2차 보고서

Github 사용 방법 소개

**9 조**

**주제** 지도 기반 채팅 커뮤니티 시스템

**팀명** CEO

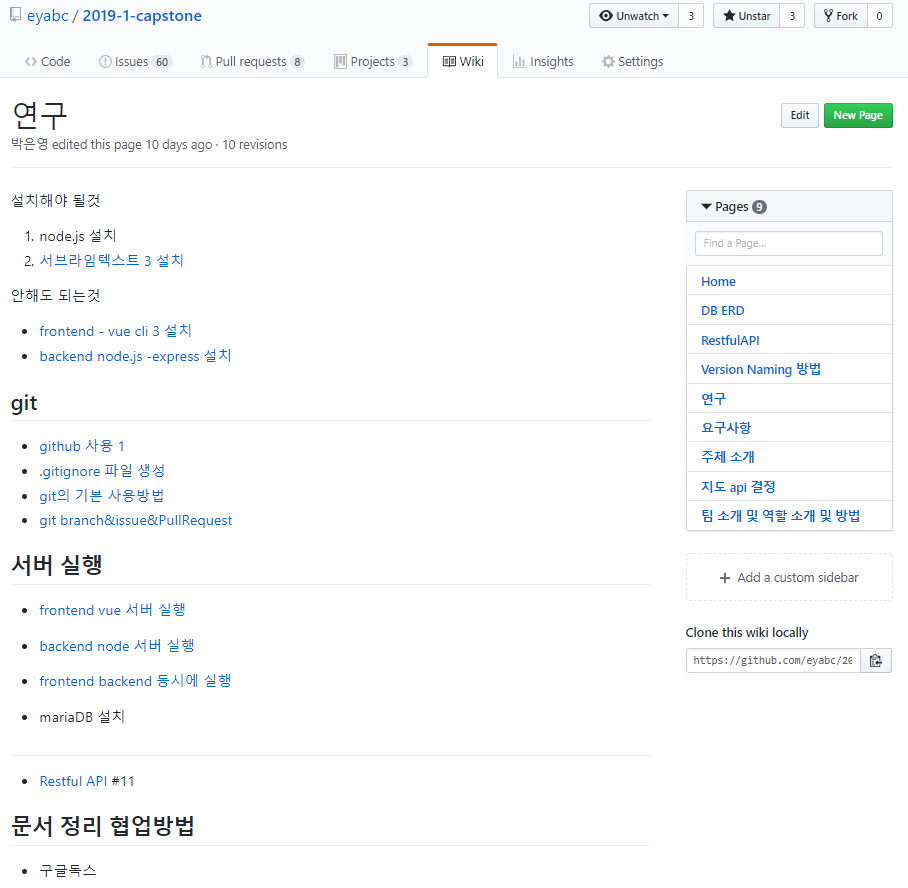
**팀원** 박은영(32151839)

**보고서 주제** github 사용방법 소개

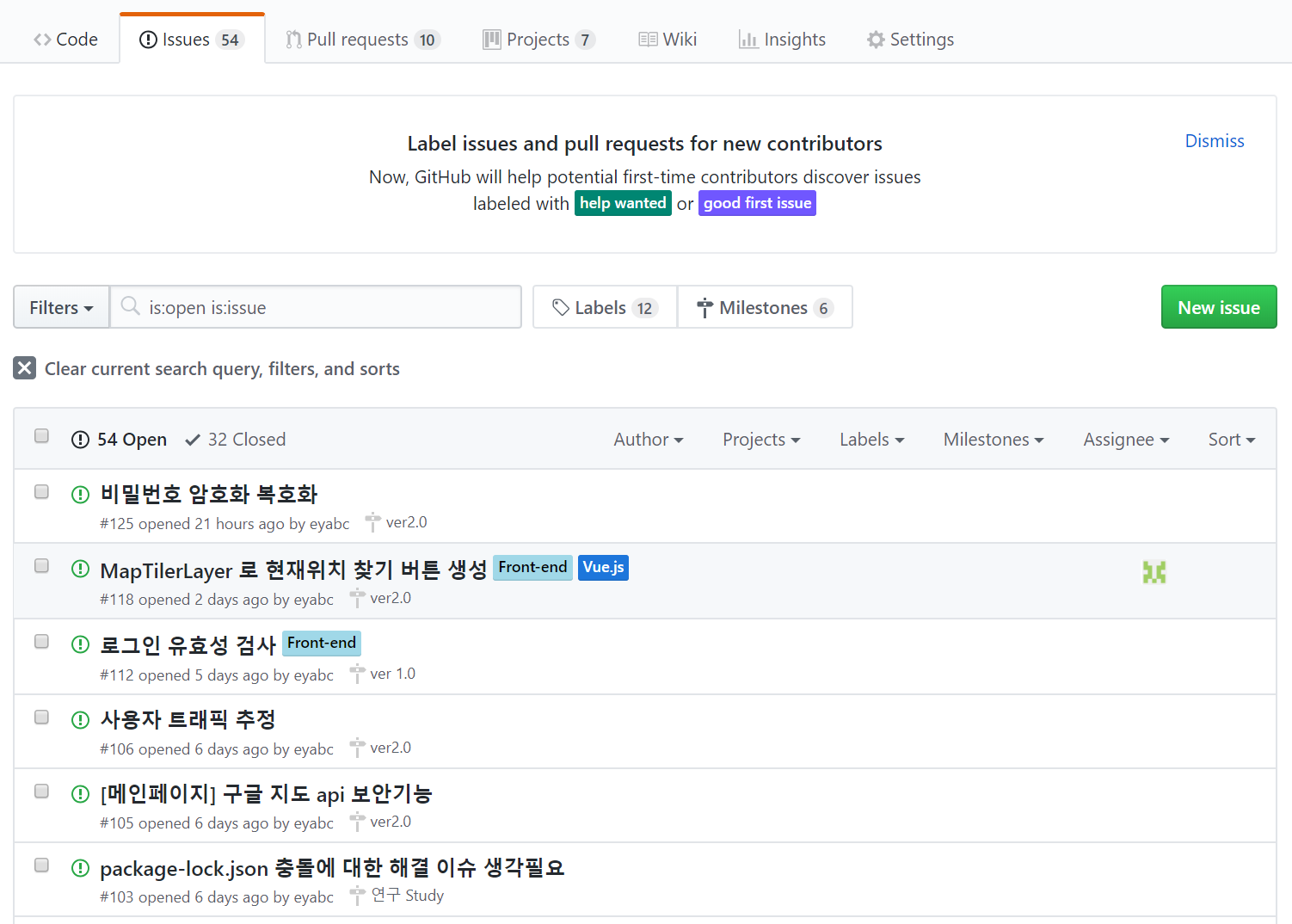
* 현재 사용중인 저장소 링크

<https://github.com/eyabc/2019-1-capstone>

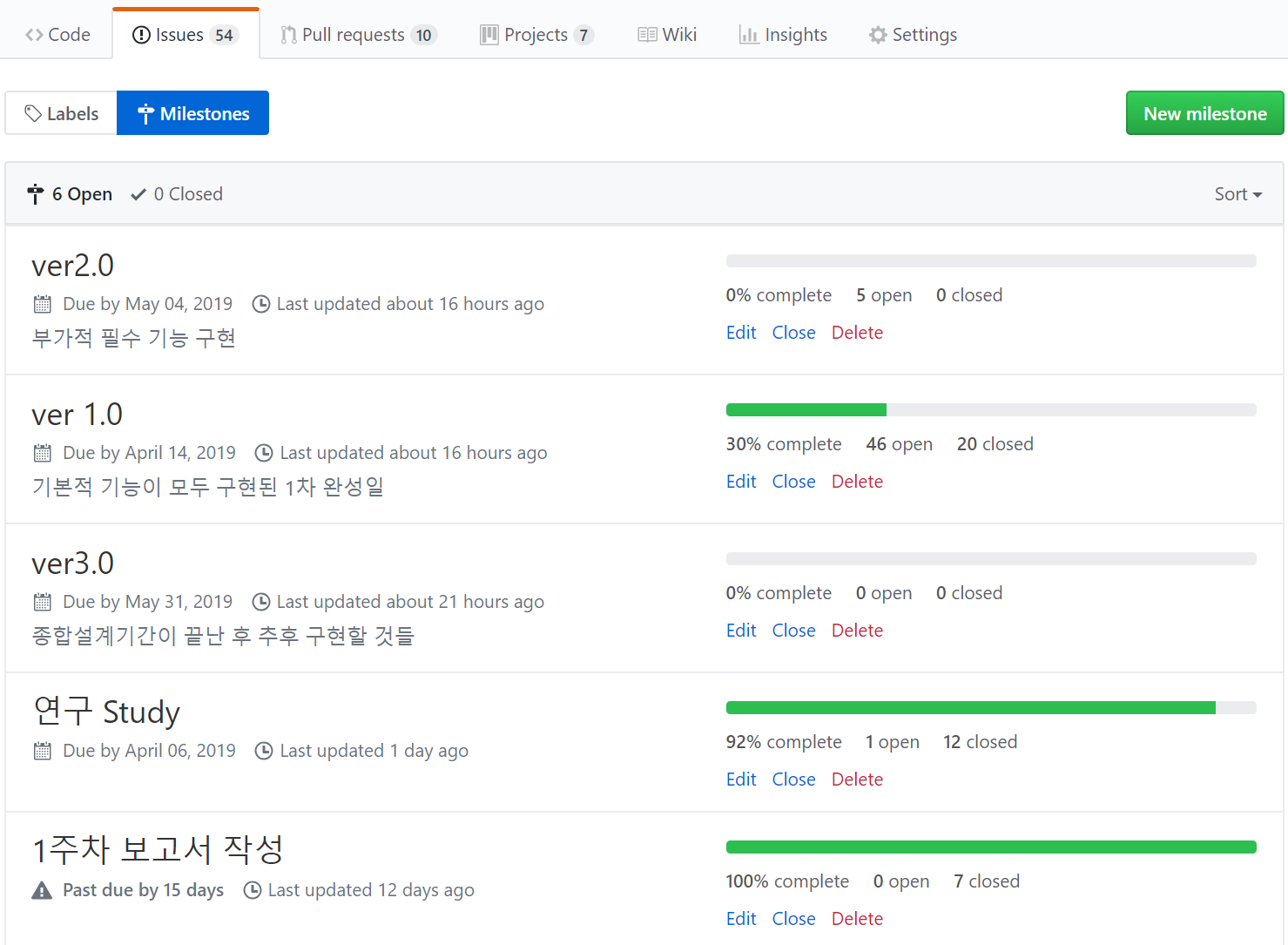
* WiKi
  1. 연구자료, RestfulAPI, 요구사항, 등 문서를 정리



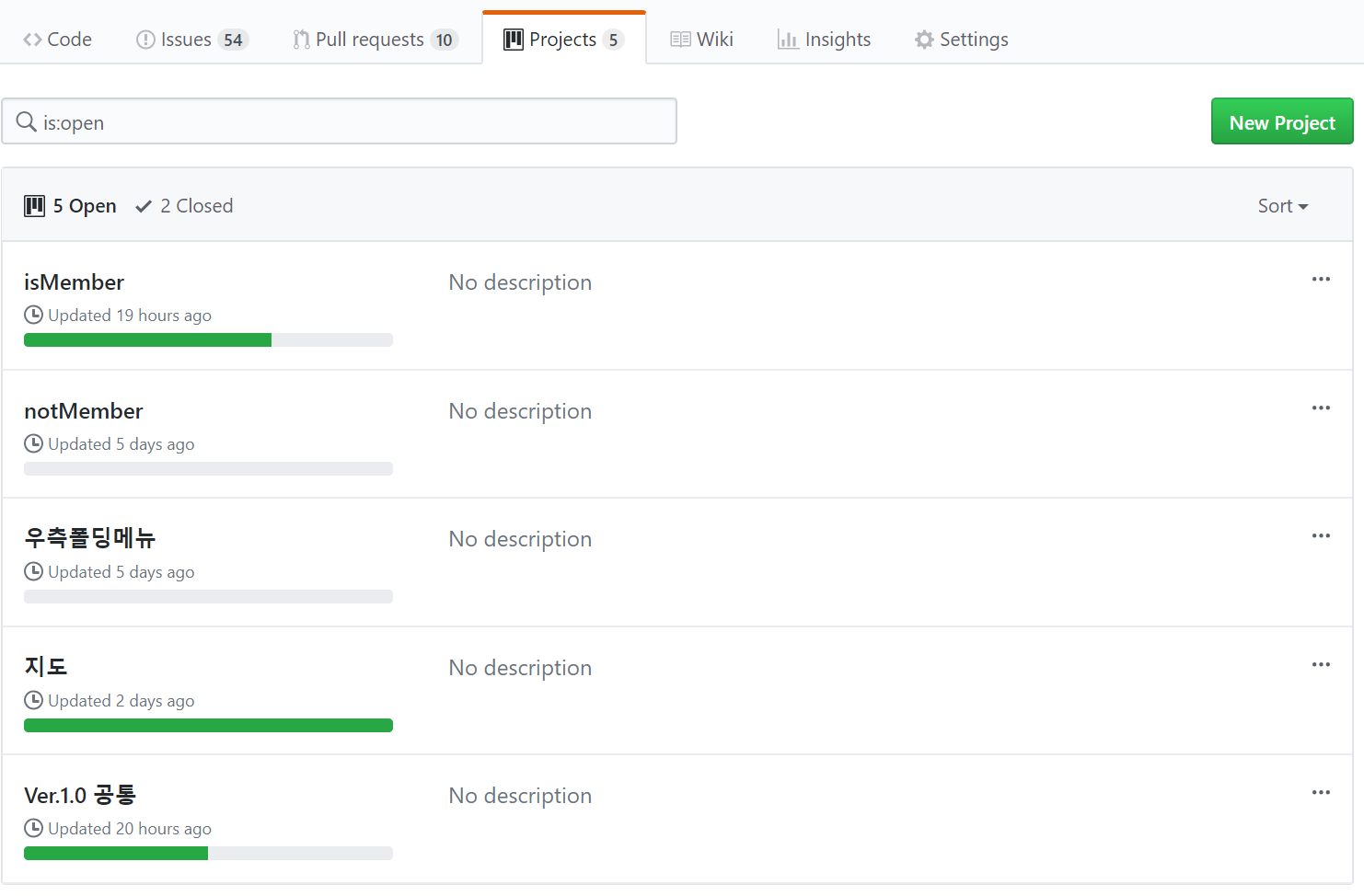
* 1. RestfulAPI, 프로토타입, 요구사항을 참조하여 이슈 탭에 구현 할 기능들을 등록했습니다.



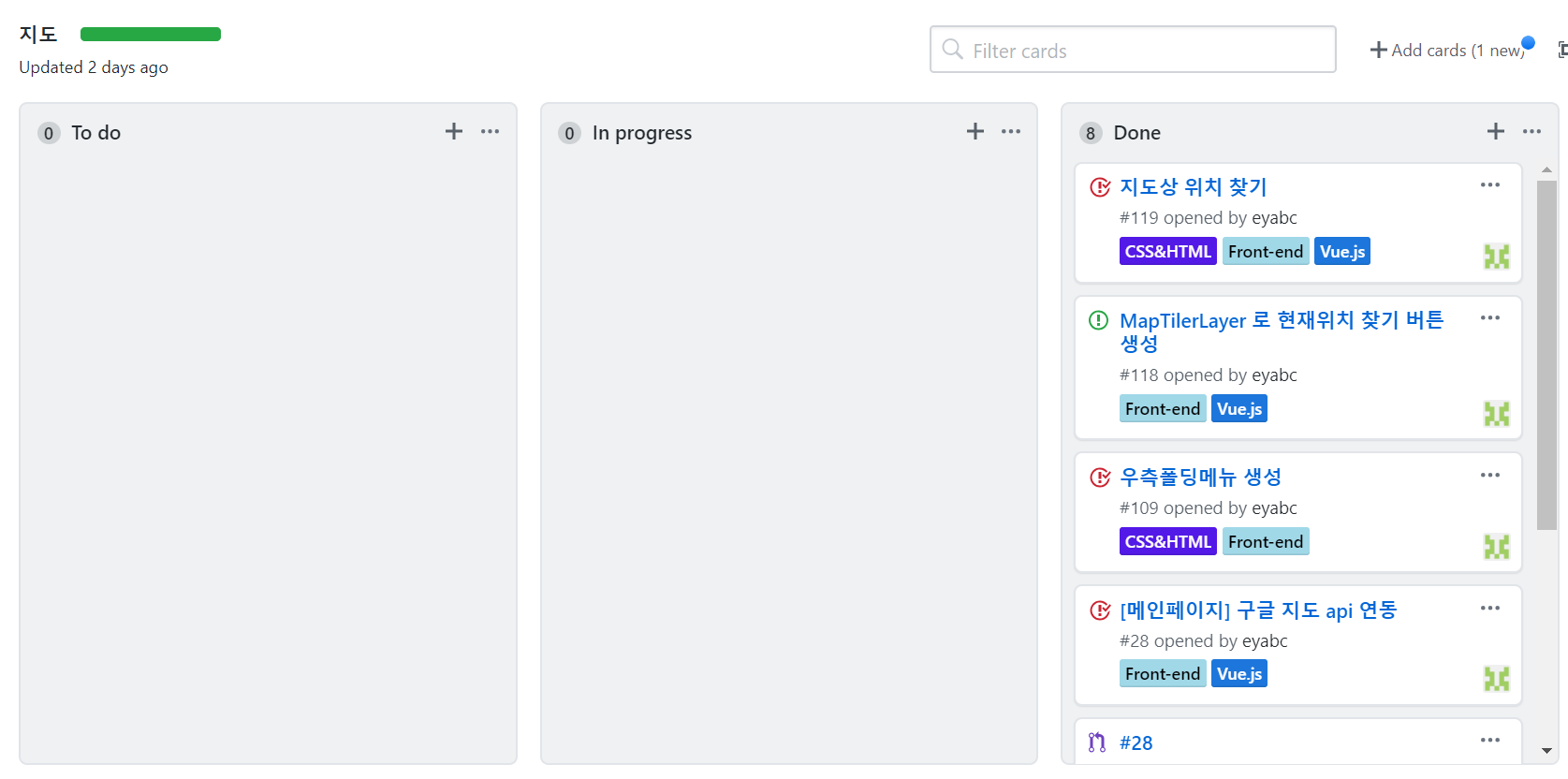
* + - 이슈 작성자, 프로젝트, 라벨, 마일스톤, 책임자 등으로 정렬 필터 가능합니다
  1. 기간의 목표를 설정하여 이슈탭의 마일스톤에 등록했습니다.



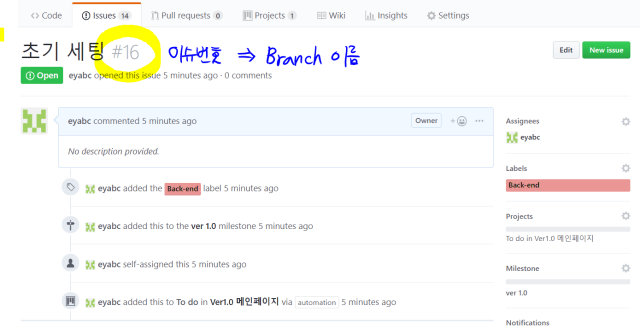
* 1. 기능별로 프로젝트 탭을 생성하였습니다.



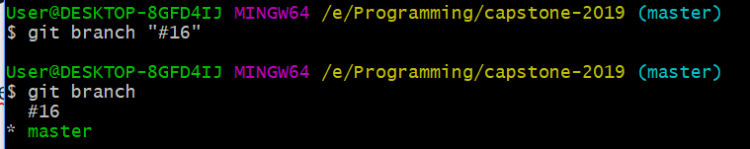
프로젝트 탭에서는 기능별 진행사항을 확인 할 수 있습니다.

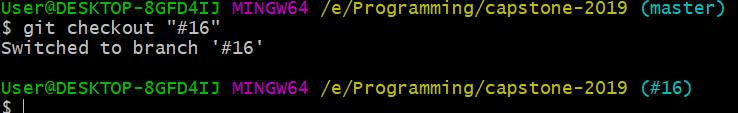


* 1. Github branch & pull Request & Issue 의 활용
     + 이슈 #16 번을 작업 하려고 합니다. Assignees에 자신을 등록합니다.



* + - 이슈번호와 동일하게 branch를 생성합니다.





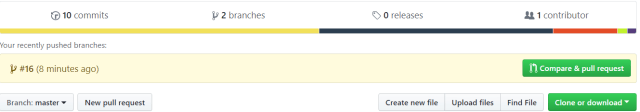
* + - 이슈에 해당하는 작업을 수행합니다

git add .

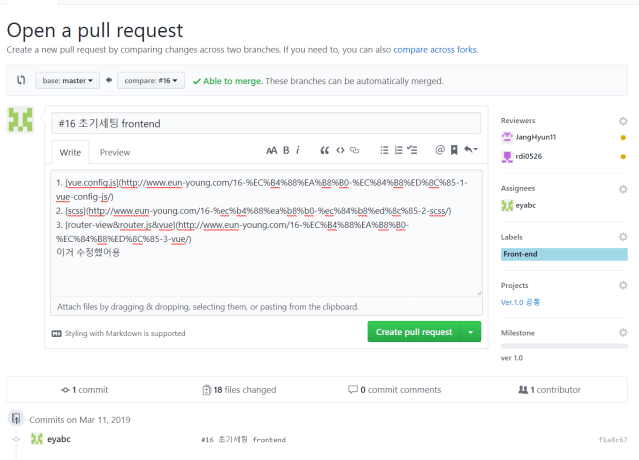
git commit -m “#16 front-end 초기세팅 완료”

git push origin “#16”

* + - push 완료 후 본인 계정의 github 저장소에 들어오면 Compare & pull reqeust 버튼이 활성화 되어 있습니다.



* + - pull Request를 작성합니다. 스크롤을 내려보면 파일이 어떻게 바뀌었는지도 조회가능합니다.

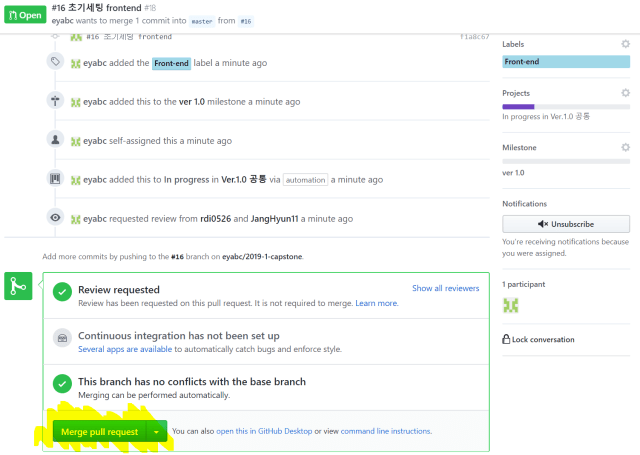


* + - 코드리뷰, Merge Pull Request

PR을 받은 Asignees는 코드 변경 내역을 확인하고 Merge를 합니다.

conflict가 있을 경우는 Merge를 할 수 없어, 실수를 방지할 수 있습니다.

또한 github 에서 conflict를 수정할 수도 있습니다.



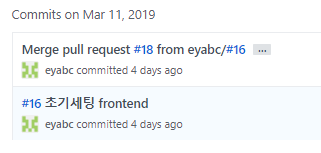
* + - Merge 할 때 세가지 옵션이 있습니다.

실험적 결과이므로 <https://meetup.toast.com/posts/122> 에서 정확한 개념을 참고해주세요

* + - * Merge pull Request

merge를 했을 때 master의 commit에 branch에서 commit 한 것이 모두 올라갑니다.

사진에서 실제로 commit을 한번 하였고. Merge pull Request를 했다는 것도 커밋이 되었습니다.



* + - * Squash and merge

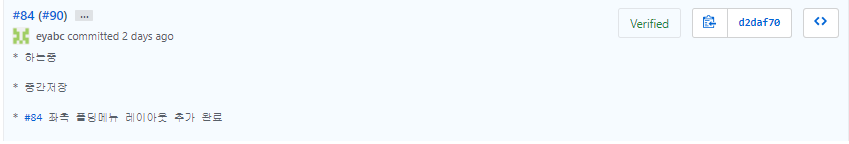
master의 commit에 branch에서 최근에 커밋한 내용 하나가 올라갑니다.

merge를 하기 전에 이전 커밋기록을 부가적으로 기록할지도 정할 수 있습니다.

… 을 눌러보면 브랜치에서 커밋한 모든 내역이 기록되어있습니다.

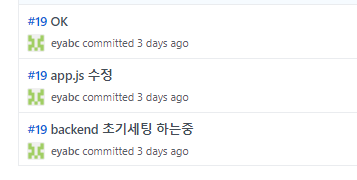
저는 이 기능을 주로 사용합니다.



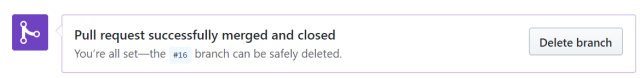
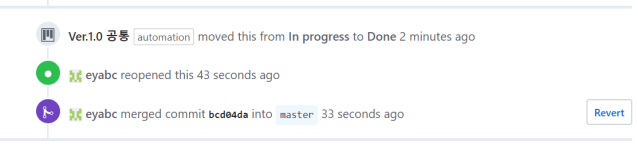


* + - * Rebase and merge

master의 commit에 branch에서 최근에 내용으로 모두 올라가고 merge했다는 commit 은 추가되지 않는다.



* + - Merge 성공! 및 확인을 합니다.

* + - 모든 작업과 테스팅이 끝나면 이슈를 close합니다.

commit을 할 때 #이슈번호를 적어주어 이슈에 commit과 merge 한것이 자동으로 기록되었습니다.

