

똑똑하게 PR을 통해 코드리뷰 하기

개발새발 · 2022년 5월 1일

팔로우

github

github



▼ 목록 보기

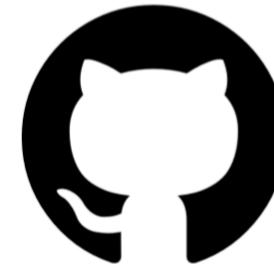
1/1



NO.54

똑똑하게 PR을 통해 코드리뷰 하기

#Github #PullRequest #PR



회사에서는 코드를 짜고 운영에 나가기 전에 Pull Request 를 통해 코드리뷰를 하고 운영에 배포하는데, 보통 코멘트를 달는 용도로만 많이 사용했습니다.

EX) 이건 이렇게 고치는게 좋을 것 같아요 ~, 이건 안맞는것 같아요 ~
요런식으로요.

그런데 웬걸 .. 어느 날 처럼 여러 깃허브를 돌아다니다가 Suggested change 라는 기능을 통해서 PR 에 코멘트를 달 수 있다는 걸 보게 되었습니다. 진짜 신선한 충격 ... 그자체 😱

바로 PR 리뷰할 때 적용하니깐 디게디게 멋있는 개발자가 된것 같고 .. PR 리뷰도 엄청 깔끔해지고 좋았습니다. 그래서 오늘은 Pull Request 의 숨겨진 기능(나만 몰랐던 기능) 들을 알아보려고 합니다.



희희

① PR 만들기

PR(풀리퀘스트)은 레파지토리 별로 확인이 가능합니다.

왼쪽에서 3번째에 있는 Pull requests 를 통해 사용할 수 있으며, New pull request 버튼으로 PR 을 만들 수 있습니다.

The screenshot shows a GitHub repository page for 'soyeon207/velog_example'. The 'Pull requests' tab is active, highlighted with a red box. Below it, there's a search bar with the query 'is:pr is:open'. A single open pull request is listed, titled 'Feat: PR 테스트 코드 추가 (#8)', which was opened 11 hours ago by 'soyeon207'. At the bottom of the page, there's a note: 'ProTip! Mix and match filters to narrow down what you're looking for.'

② PR 구조 살펴보기

풀리퀘스트를 만들게 되면 아래와 같은 모습이 됩니다.

하나하나씩 뜯어보면서 구조를 살펴보도록 하겠습니다.

Feat: PR 테스트 코드 추가 (#8) #9

The screenshot shows a GitHub Pull Request page for a pull request titled "Feat: PR 테스트 코드 추가 (#8) #9". The description field contains the text "No description provided.". To the right of the description, there are several status indicators and settings: "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), "Development" (None yet), and "Notifications" (Customize, Unsubscribe). Below the description, there is a comment from "soyeon207" stating "No description provided." and a note about continuous integration.

②-1. description

description 은 풀리퀘스트 관련 설명을 쓸 수 있는 공간입니다.
오른쪽 위에 있는 . . . 을 통해 Edit 을 할 수 있습니다.

The screenshot shows the same GitHub Pull Request page as before, but with a context menu open over the "No description provided." text in the description field. The menu options are "Copy link", "Quote reply", "Edit" (which is highlighted in blue), and "Report content".

내용은 마크다운으로 작성하면 됩니다.

The screenshot shows the GitHub rich text editor interface. The "Write" tab is selected, and the "Preview" tab is visible above the editor area. The editor content is as follows:

```
### PR 테스트 코드 추가
어떤거어떤거 확인 부탁 드립니다
```

Below the editor, there is a note: "Attach files by dragging & dropping, selecting or pasting them." At the bottom right of the editor are "Cancel" and "Update comment" buttons.

②-2. Reviewers & Assigners

오른쪽 상단위에 있는 Reviewers 에는 리뷰 해줄 사람을 Assignees 에는 PR 을 작업한 담당자를 지정해주면 됩니다.
반드시 한 사람만 지정하는게 아니라 여러 사람을 지정할 수 있습니다.

Reviewers



No reviews

Still in progress? Convert to draft

Assignees



soyeon207

②-3. 리뷰 작성하기

리뷰는 상단에 있는 Files changed 를 통해 남길 수 있습니다.

한줄 혹은 여러줄에 리뷰를 남길 수 있습니다.

한줄에 남기기

남기고자 하는 줄의 왼쪽에 마우스를 가져가면 + 버튼이 보이는데 이걸 누르면 해당 줄에 대해 리뷰를 남길 수 있습니다.

```

v 18 README.md
...
...
@@ -1,11 +1,11 @@
1 1 # velog_example
2 + velog 에 포스팅 하는 예제 코드입니다.
2 + velog 에 포스팅 하는 예제 코드 가 아닙니다.
3 
4 - soyeon207 velog - https://velog.io/@soyeon207
5 -
6 - |블로그 글 |모듈 이름|포트|
4 + |블로그 글 |모듈 이름|포트|
7 5 |---|---|---|
8 6 |[Swagger 란 ? {이론 + Spring boot 적용}]|(https://velog.io/@soyeon207/%EC%9A%8B%EB%BB%9ED%83%95%ED%83%95-Swagger-%EC%8A%81%EC%9A%9%EA%BB%80) |swagger|
9 7 |[[설명판] 스프링 시큐리티 적용하기](https://velog.io/@soyeon207/%EC%8B%A4%EC%8A%85%ED%8E%88-%EC%8A%A4%ED%94%84%EB%A7%81-%EC%BB%9C%ED%81%9%EB%8A%6%AC%ED%81%
NEC%AB%81%EC%9A%9%ED%95%98%EA%BB%80)|spring-security-server|8802|
10 8 |[[설명판] Spring Boot에서 Spring Security 기반 JWT 적용하기](https://velog.io/@soyeon207/JWT-%EC%8B%A4%EC%8A%85) |jwt-security-server|8001|
11 - |ooo|es-test-server|8003
9 + |ooo|es-test-server|8003|
10 +
11 + 테스트테스트

```

여러줄에 남기기

한줄 남기기처럼 + 버튼이 보이면 남기고자 하는 줄까지 아래로 땅겨주면 여러 줄에 대해 리뷰를 남길 수 있습니다.

```

v 18 README.md
...
...
@@ -1,11 +1,11 @@
1 1 # velog_example
2 - velog 에 포스팅 하는 예제 코드입니다.
2 + velog 에 포스팅 하는 예제 코드 가 아닙니다.
3 
4 + soyeon207 velog - https://velog.io/@soyeon207
5 -
6 - |블로그 글 |모듈 이름|포트|
4 + |블로그 글 |모듈 이름|포트|
7 5 |---|---|---|
8 6 |[Swagger 란 ? {이론 + Spring boot 적용}]|(https://velog.io/@soyeon207/%EC%9A%8B%EB%BB%9ED%83%95%ED%83%95-Swagger-%EC%8A%81%EC%9A%9%EA%BB%80) |swagger|
9 7 |[[설명판] 스프링 시큐리티 적용하기](https://velog.io/@soyeon207/%EC%8B%A4%EC%8A%85%ED%8E%88-%EC%8A%A4%ED%94%84%EB%A7%81-%EC%BB%9C%ED%81%9%EB%8A%6%AC%ED%81%
NEC%AB%81%EC%9A%9%ED%95%98%EA%BB%80)|spring-security-server|8802|
10 8 |[[설명판] Spring Boot에서 Spring Security 기반 JWT 적용하기](https://velog.io/@soyeon207/JWT-%EC%8B%A4%EC%8A%85) |jwt-security-server|8001|
11 - |ooo|es-test-server|8003
9 + |ooo|es-test-server|8003|
10 +
11 + 테스트테스트

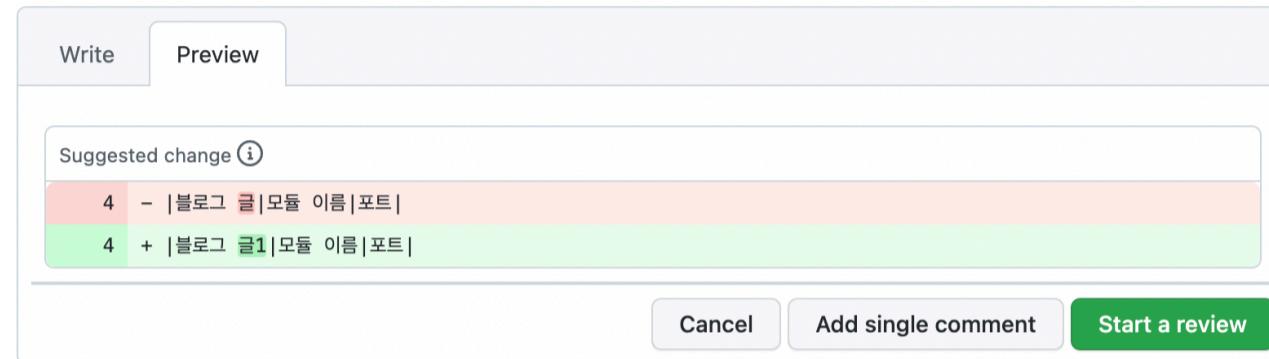
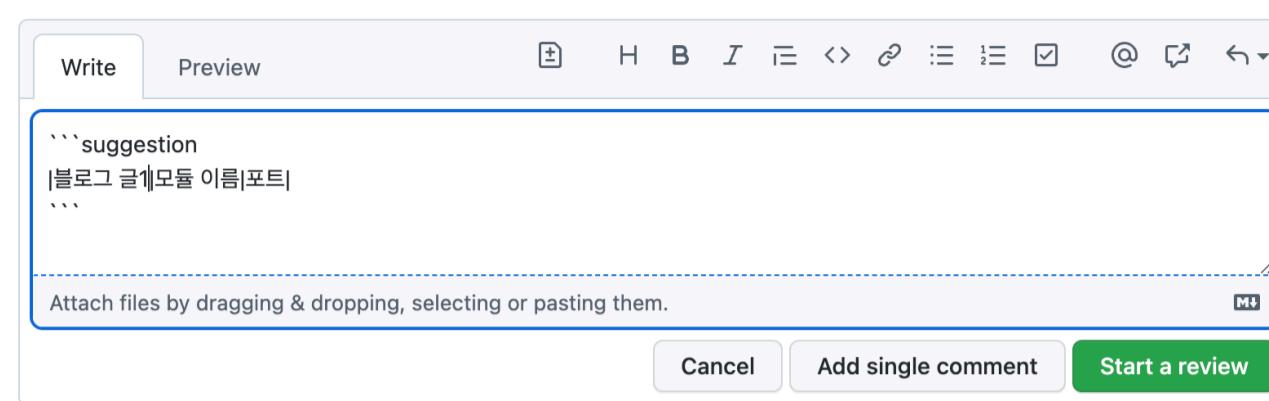
```

리뷰를 남기기 위한 창은 다음과 같이 구성되어 있습니다.

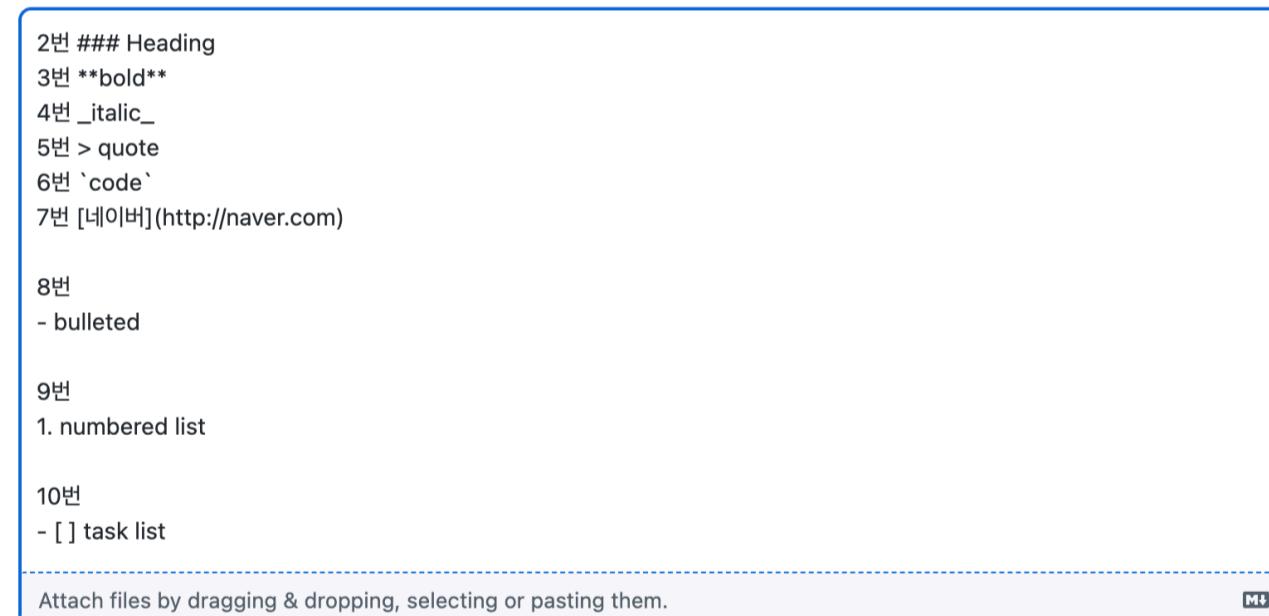
The screenshot shows the 'Write' tab selected in the top navigation bar. Below it is a large text area with a placeholder 'Leave a comment'. At the bottom of this area is a dashed line with the text 'Attach files by dragging & dropping, selecting or pasting them.' To the right of the text area are several icons for rich text editing: bold (B), italic (I), underline (U), strikethrough (S), numbered list (1-13), and other common document tools. At the very bottom are three buttons: 'Cancel', 'Add single comment', and a green 'Start a review' button.

1번 Add a suggestion

제가 이번에 알고 가장 놀랐던 기능인데요 ..
 해당 코드에 대해서 제안할 수 있는 기능입니다.
 어떤 부분이 어떻게 바뀌었는지도 함께 보여줍니다.

**2번 ~ 10번** 은 마크다운 문법입니다.

버튼을 눌러서 사용하고 평소 마크다운 언어를 사용하는 것처럼 사용해도 됩니다.
 아래와 같이 작성되며,



이걸 Preview로 보면 아래와 같이 보이게 됩니다.

The screenshot shows a 'Write' tab selected, displaying a preview of the rich text editor. The text area contains numbered and bulleted lists, along with various styling options like bold and italic. Below the text area are three buttons: 'Cancel', 'Add single comment', and a green 'Start a review' button.

11번 과 **12번** 은 각각 멘션 기능과 이슈 혹은 풀리퀘에 대한 바로 가기 링크입니다.
아래처럼 Write 할 수 있고,

The screenshot shows a 'Write' tab selected, displaying a preview of the rich text editor. The text area contains mentions (@soyeon207), issue links (#8), and timeline links (#9). Below the text area are three buttons: 'Cancel', 'Add single comment', and a green 'Start a review' button.

Preview로 보게 되면 아래처럼 보여지게 됩니다.

The screenshot shows a 'Preview' tab selected, displaying the rendered preview of the rich text editor. It includes the mentions (@soyeon207), issue links (#8), and timeline links (#9) from the previous screenshot. Below the preview area are three buttons: 'Cancel', 'Add single comment', and a green 'Start a review' button.

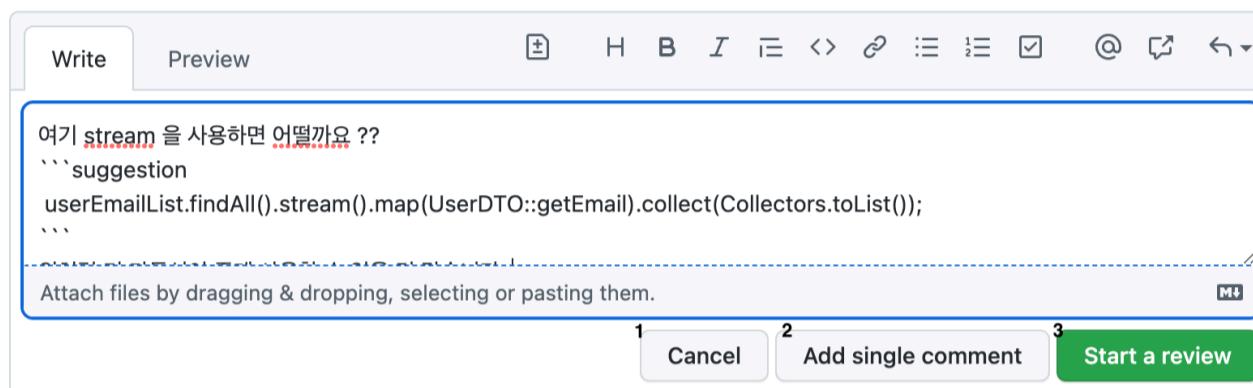
13번 의 경우에는 자동완성 기능입니다.

누르게 되면 Select a reply 라는 글이 보이게 되는데요. 여기에 보여진 답장들은 settings에서 설정할 수 있습니다.

The screenshot shows a 'Write' tab selected, displaying a preview of the rich text editor. A dropdown menu titled 'Select a reply' is open, listing several saved replies: 'LGTM' (ctrl 1), 'Default replies', 'Duplicate pull request' (ctrl 2), and 'Create a new saved reply...'. Below the preview area are three buttons: 'Cancel', 'Add single comment', and a green 'Start a review' button.

Code, planning, and automation > Saved replies에서
타이틀 및 내용을 지정해 저장할 수 있고, 저장된 reply는 13번을 선택했을 때 노출되게 됩니다.

②-4. 리뷰 적용하기



리뷰를 모두 작성하고 남기려고 하는데 오잉 ? 아래에 총 3가지의 버튼이 있습니다.

어떤 버튼을 누르면 되는걸까요 ??

첫번째, Cancel

해당 리뷰를 남기지 않겠다는 취소 버튼입니다.

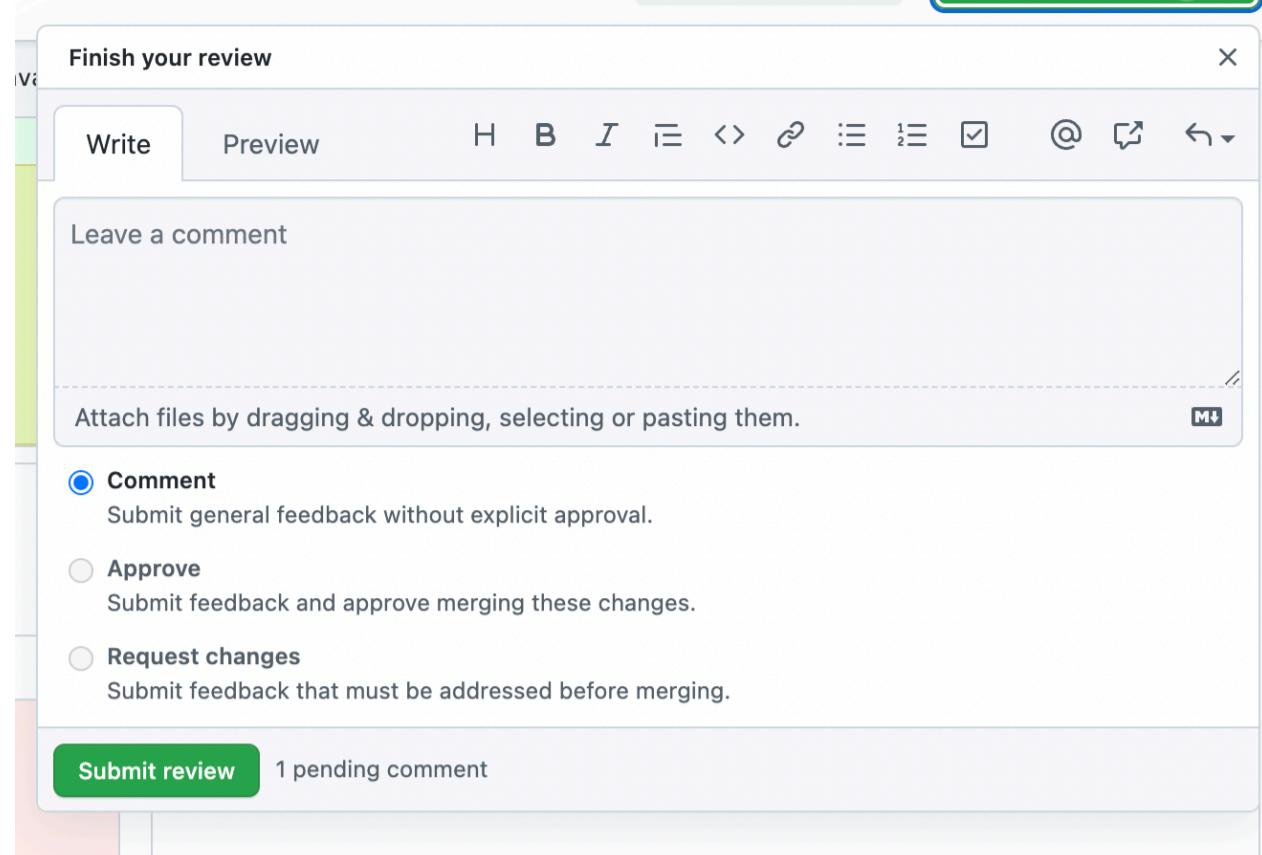
두번째, Add single comment

바로 리뷰를 등록하는 버튼입니다.

세번째, Start a review

두번째와 다르게 바로 리뷰를 등록하지 않고, pending 상태로 등록하고 한번에 리뷰를 남길 수 있는 버튼입니다.

세번째 버튼으로 리뷰를 남기게 되면 리뷰 옆에 Pending 이라는 상태가 붙게 되고, 리뷰 페이지의 가장 위쪽 오른쪽에 Finish your review (숫자) 버튼이 활성화 되게 됩니다.



여기도 세가지의 상태를 선택할 수 있는데요

첫번째, Comment

일반적인 코멘트입니다.

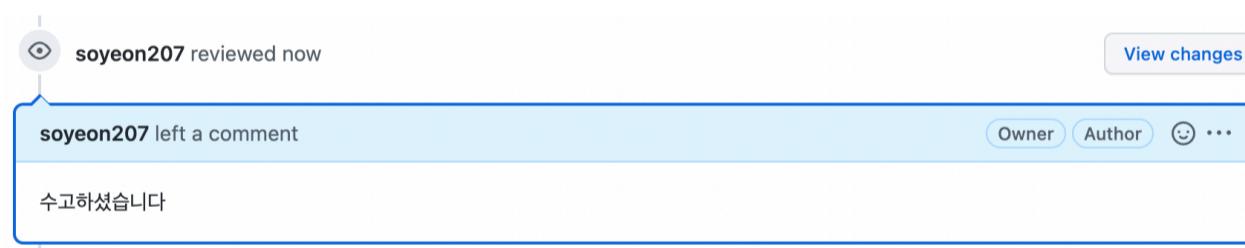
두번째, Approve

리뷰어가 승인하는 것으로 머지해도 괜찮다는 것입니다.

세번째, Request changes

리뷰어가 승인하지 않는 것으로 수정 해달라는 요청입니다.

Finish your review 에 코멘트를 남기게 되면 아래처럼 보이게 됩니다.



②-5. 추가 설정

Approve 가 되어야지 머지가 되도록 설정할 수도 있습니다.

설정하고 싶은 Repository > Settings 을 선택하고,
Code and automation > Branches 메뉴를 눌러줍니다.

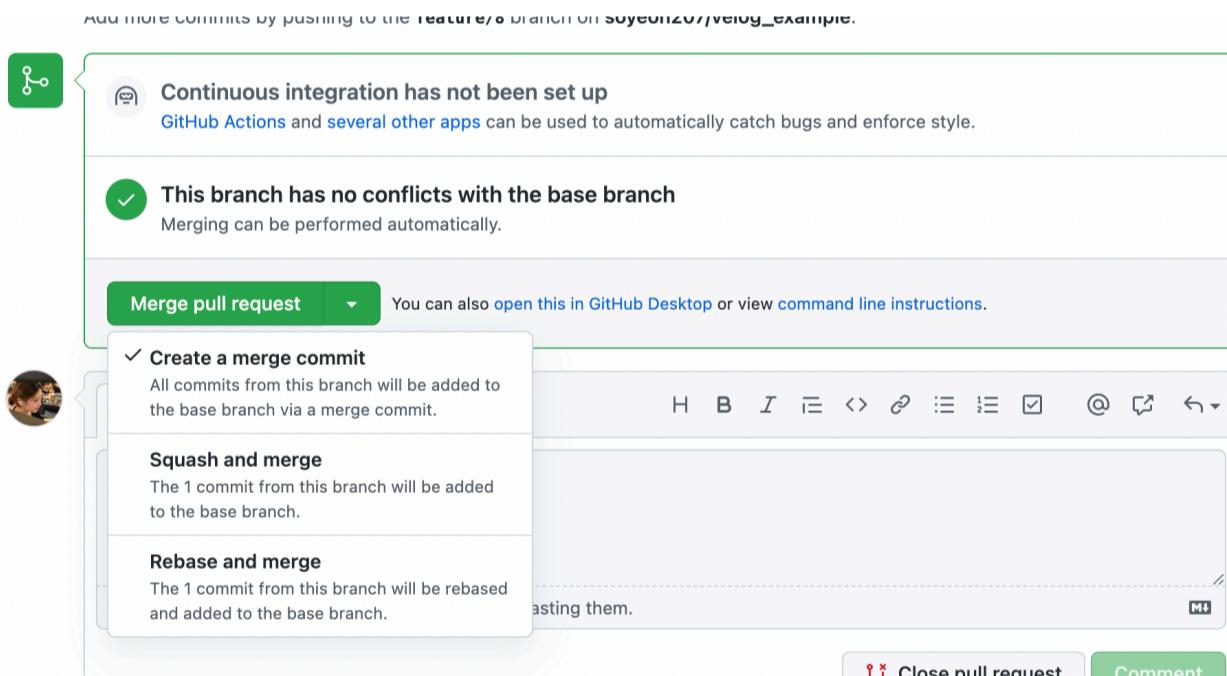
그러면 Protect matching Branches에서 Require a pull request before merging이라는 설정을 통해 Approve가 되어야지 머지가 되도록 설정할 수 있습니다.

②-6. 머지하기

모든 리뷰가 끝났다 ~~

라고 하면 이제 머지를 해야겠죠? 😊

머지는 Conversation의 아래쪽에 있는 버튼을 통해 머지를 할 수 있습니다.



근데 여기도 설정할 수 있는게 총 3가지가 있습니다.

첫번째, Create a merge commit

PR의 commit들이 main 브랜치의 HEAD commit으로 들어갑니다.

git merge 명령어와 같으며, 메시지는 수정할 수 없습니다.

브랜치의 커밋 기록들이 직관적으로 병합된다는 장점이 있습니다.

두번째, Squash and merge

PR의 commit을 한 개로 추려서 main 브랜치에 병합됩니다.

PR 제목으로 된 1개의 로그로 main에 병합됩니다.

세번째, Rebase and merge

PR의 커밋 로그들이 main에 재정렬되어 병합됩니다.

commit 기록이 구구절절 main에 한줄로 되어 마치 main에서 다 작업한 것처럼 로그가 남게 됩니다.

해당 글을 참고해서 앞으로 PR에 리뷰를 남길 때 조금 더 똑똑하게 남길 수 있기를 ... 바랍니다



레퍼런스

[Github] Pull Request를 통해 코드리뷰(Code Review)하는 법

[Git/GitHub] 3. Pull Request (Collaborator 추가)