

8. Github 알아보기와 실습

- 1. Github Docs
- 2. Local Git 과 Github Repository 연동
- 3. PR 활용하기
 - PR 템플릿 만들기
 - PR 생성하기
- 4. Github Page
- 5. Github Release
- 6. Github Package

1. Github Docs

 [GitHub.com 도움말 문서](#)

Get started

[시작](#)
[마이그레이션](#)
[계정 및 프로필](#)
[인증](#)
[청구 및 결제](#)
[사이트 정책](#)

Collaborative coding

[GitHub Codespaces](#)
[리포지토리](#)
[끌어오기 요청](#)
[GitHub 토론](#)
[GitHub Copilot](#)

CI/CD and DevOps

[GitHub 작업](#)
[GitHub 패키지](#)
[GitHub 페이지](#)

Security

[코드 보안](#)
[공급망 보안](#)
[보안 공지](#)
[Dependabot](#)
[코드 검사](#)
[비밀 검사](#)

Client apps

[GitHub CLI](#)
[GitHub Desktop](#)

Project management

[GitHub 문제](#)
[Projects를 사용하여 계획 및 추적](#)
[작업 목록](#)
[GitHub에서 검색](#)

Developers

[업](#)
[REST API](#)
[GraphQL API](#)
[웹훅크](#)

Enterprise and Teams

[조직](#)
[엔터프라이즈 관리자](#)

Community

[커뮤니티 빌드](#)
[GitHub 스폰서](#)
[Education](#)
[GitHub 지원](#)
[GitHub Docs에 기여](#)

More docs

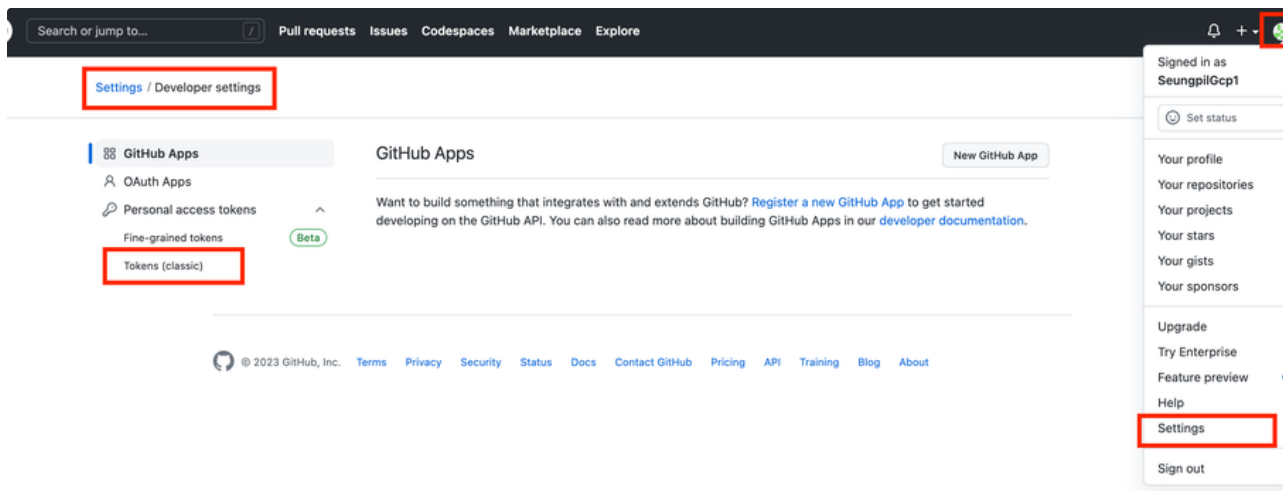
[Electron !\[\]\(9f3852d68d41e1e95bc4ec10e81aba4b_img.jpg\)](#)
[CodeQL !\[\]\(bdb0f3befa0da67ab318d6edc913daee_img.jpg\)](#)
[npm !\[\]\(bf8000641e0d351c7a9edc2d7fb94a96_img.jpg\)](#)

2. Local Git 과 Github Repository 연동

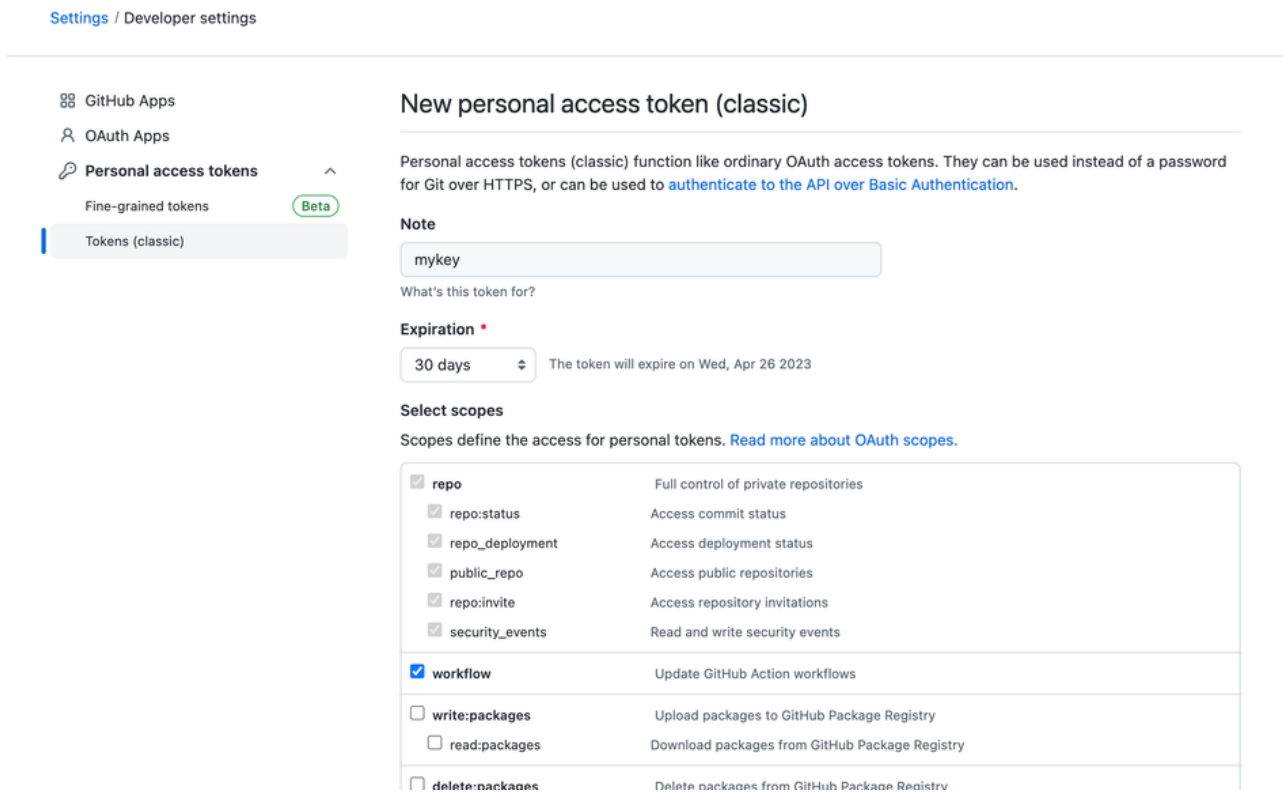
Github는 Git을 기반으로 한 코드 호스팅 플랫폼입니다. 이는 개발자들에게 코드 관리 및 협업을 쉽게 할 수 있는 기능을 제공합니다.

실습은 여전히 CodeSpace 기반으로 진행합니다. 실제로 Local Git 에서 Github Repository 를 Remote 저장소로 사용하고 싶으시다면 다음 절차로 Token 을 발급받아 진행할 수 있습니다.

- 우측 상단 프로필 > Settings > Developer settings > Tokens



- 토큰 이름과 권한 설정



- 생성된 토큰 저장

- GitHub Apps
- OAuth Apps
- Personal access tokens
 - Fine-grained tokens (Beta)
 - Tokens (classic)

Personal access tokens (classic)

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

Delete

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](#).

- Repository > Code > Https 주소 git clone

SeungpilGcp1 / day1 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file <> Code

SeungpilGcp1 made other changes

CONTRIBUTING.md	added new benchmarks
README.md	add new file
new_file	Create new file
new_file2	add new file
test.rb	made other changes

README.md

day1

Local Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/SeungpilGcp1/day1.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

- Repository local git 과 연동

```

1 # 사용자 세팅
2 $ git config --global user.name "SeungpilGcp1"
3 $ git config --global user.email "seungpilgcp1@gmail.com"
4
5 # Repository 내려받고 token 설정하기
6 $ git clone https://github.com/SeungpilGcp1/day1
7 $ git remote remove origin
8 $ git remote add origin https://<token>@github.com/SeungpilGcp1/day1
9
10 $ git push --set-upstream origin main
11 Enumerating objects: 5, done.
```

```

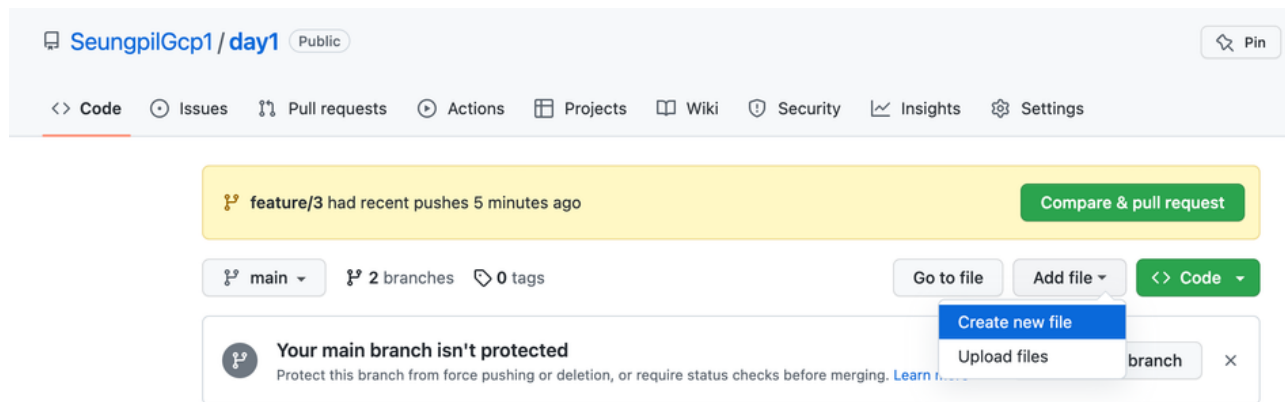
12 Counting objects: 100% (5/5), done.
13 Delta compression using up to 10 threads
14 Compressing objects: 100% (2/2), done.
15 Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
16 Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
17 remote: Resolving deltas: 100% (1/1), completed with 1 local object.
18 To https://github.com/SeungpilGcp1/day1
19   c64cc2e..65c5792  main -> main
20 Branch 'main' set up to track remote branch 'main' from 'origin'.

```

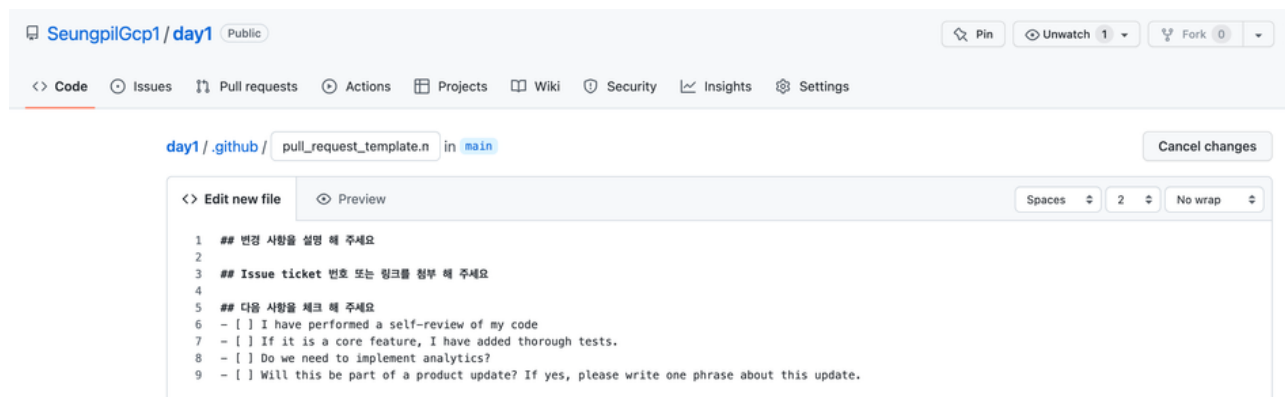
3. PR 활용하기 ↗

PR 템플릿 만들기 ↗

- Code > create new file



- .github/pull_request_template.md 파일 생성



```

1 ## 변경 사항을 설명 해 주세요
2
3 ## Issue ticket 번호 또는 링크를 첨부 해 주세요
4
5 ## 다음 사항을 체크 해 주세요
6 - [ ] I have performed a self-review of my code
7 - [ ] If it is a core feature, I have added thorough tests.

```

```
8 - [ ] Do we need to implement analytics?
9 - [ ] Will this be part of a product update?
```

PR 생성하기

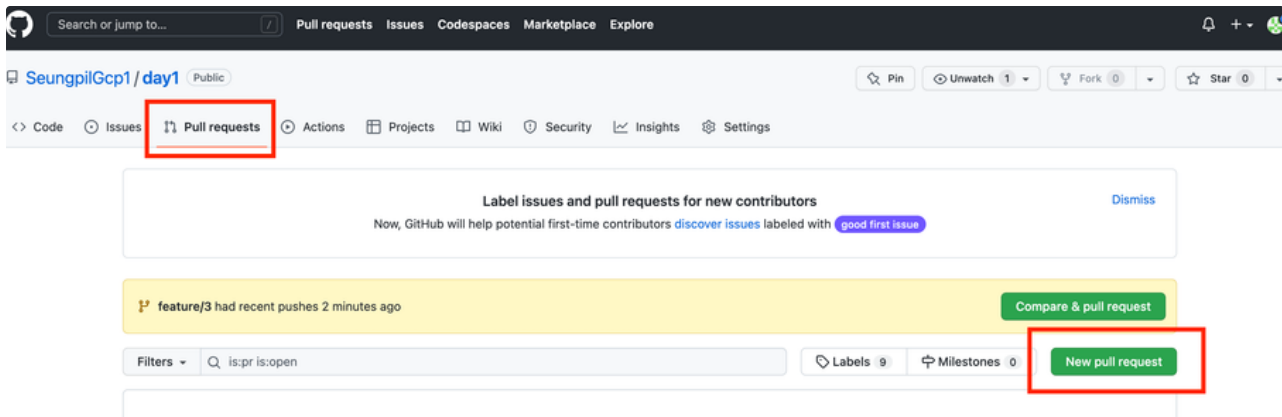
- 그림처럼 feature/3 브랜치를 생성후, index.html 을 만들어 origin 에 push

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hello, World!</title>
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>
```

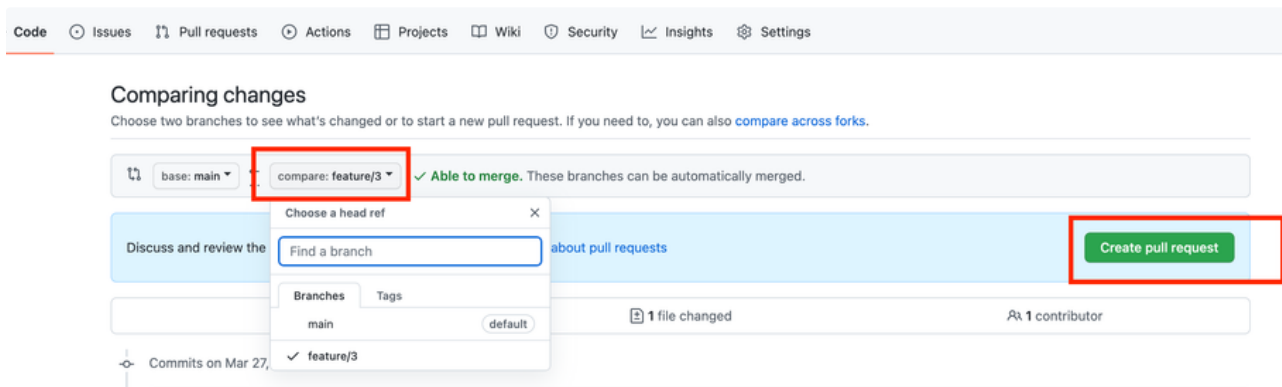
The screenshot shows the VS Code interface with the following components:

- Commit Graph: day1**: Shows a single commit on the 'feature/3' branch.
- File Explorer**: Lists files in the 'DAY1 [CODESPACES]' workspace, including 'index.html' which is currently selected.
- Editor**: Displays the content of 'index.html', which is a simple HTML document with a title 'Hello, World!' and a body containing 'Hello, World!'.
- Terminal**: Shows the output of a series of Git commands executed in the workspace. The commands include checking out a new branch 'feature/3', adding the 'index.html' file, committing the changes, and pushing them to the remote repository. The output shows the successful creation of the branch and the push operation.

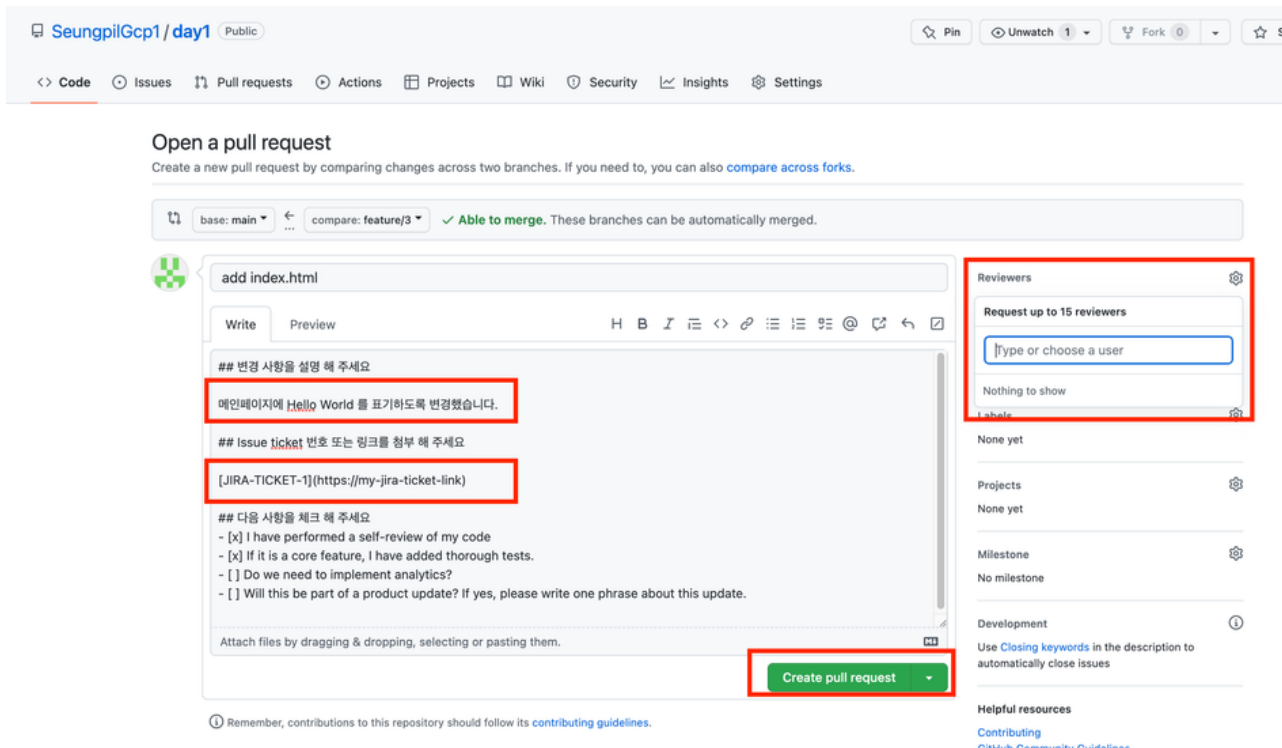
- Github > Pull requests > 신규 PR 생성



- merge 될 브랜치 선택



- 템플릿이 나타나게 되고, 텍스트 작성
- 코드 리뷰를 받을 인원 추가



- Confirm 시 Merge 되면서 PR 은 종료상태가 됨.

add index.html #1

Merged SeungpilGcp1 merged 1 commit into `main` from `feature/3` [now](#)

Conversation 0 Commits 1 Checks 0 Files changed 1

SeungpilGcp1 commented 4 minutes ago Owner ...

변경 사항을 설명 해 주세요


메인페이지에 Hello World 를 표기하도록 변경했습니다.

Issue ticket 번호 또는 링크를 첨부 해 주세요

[JIRA-TICKET-1](#)

다음 사항을 체크 해 주세요

- ☒ I have performed a self-review of my code
- ☒ If it is a core feature, I have added thorough tests.
- ☐ Do we need to implement analytics?
- ☐ Will this be part of a product update? If yes, please write one phrase about this update.

 `add index.html` 0ba9d5f

SeungpilGcp1 merged commit `c52f538` into `main` [now](#) Revert

Pull request successfully merged and closed Delete branch

You're all set—the `feature/3` branch can be safely deleted.

4. Github Page [🔗](#)

- 이전 PR 에서 index.html 을 main 에 병합 후
- Github Pages 를 통해 static web site 를 구현할 수 있다.
- Settings > Pages > branch 선택 후 save

SeungpilGcp1 / day1 (Public)

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub apps

Email notifications

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

main / (root) Save

Visibility GITHUB ENTERPRISE

With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. A privately published site can only be accessed by people with read access to the repository the site is published from. You can use privately published sites to share your internal documentation or knowledge base with members of your enterprise.

[Try GitHub Enterprise risk-free for 30 days](#) [Learn more about the visibility of your GitHub Pages site](#)

- Github Actions 에서 build 수행 확인 > deploy 결과의 URL 클릭시

SeungpilGcp1 / day1 (Public)

Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

pages build and deployment #1

Summary

Jobs

- build
- report-build-status
- deploy

Run details

Usage

Triggered via dynamic 47 minutes ago

Status: **Success**

Total duration: **53s**

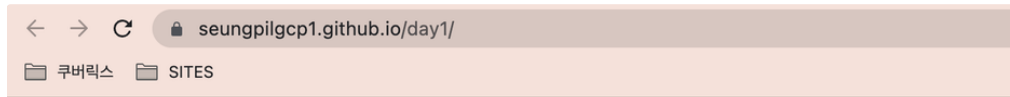
Artifacts: **1**

pages-build-deployment
on: dynamic

```

graph LR
    build[build 20s] --> report[report-build-status 2s]
    report --> deploy[deploy 7s]
    deploy --> url[https://seungpilgcp1.github.io/day1/]
  
```


- 인터넷 상에서 접속가능한 페이지 확인

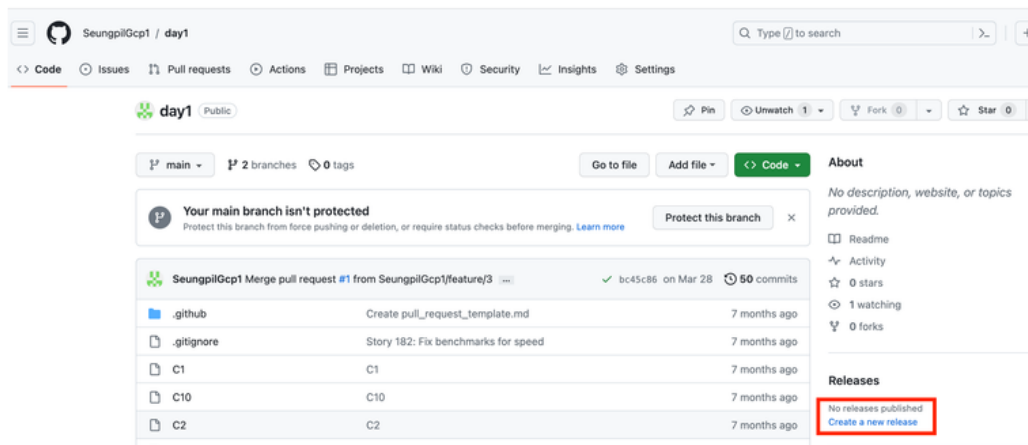


Hello, World!

5. Github Release [↗](#)

github 에 Prjecct 를 진행하고 있었고 목표한 기능에 도달했다면 다음 절차에 따라 Release 를 할 수 있고 이때 빌드된 바이너리 파일도 같이 업로드 할 수 있다.

- Create a new release 버튼 클릭



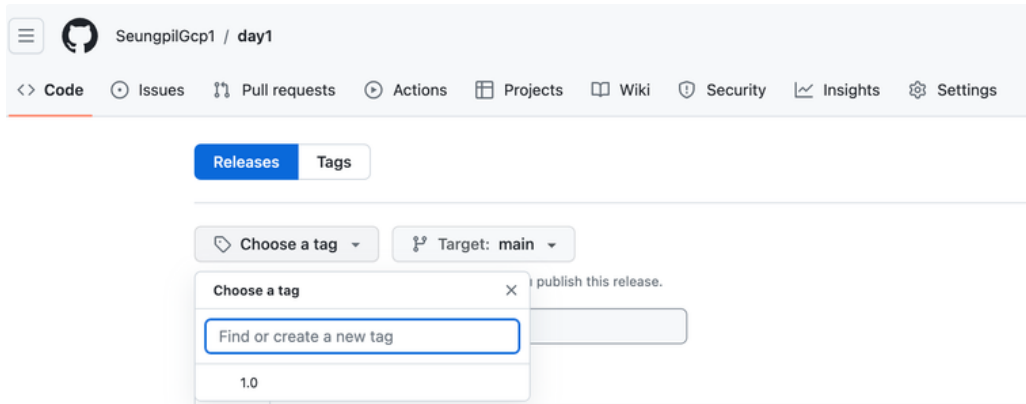
1. 태그를 설정해야 한다.

깃허브의 릴리즈는 저장소(Repository)의 특정 지점 기록을 표시하는 [Git Tag](#) 를 기반으로 작동하게 된다.

여기서 말하는 태그란 쉽게 말해, **릴리즈의 버전 번호**를 의미한다.

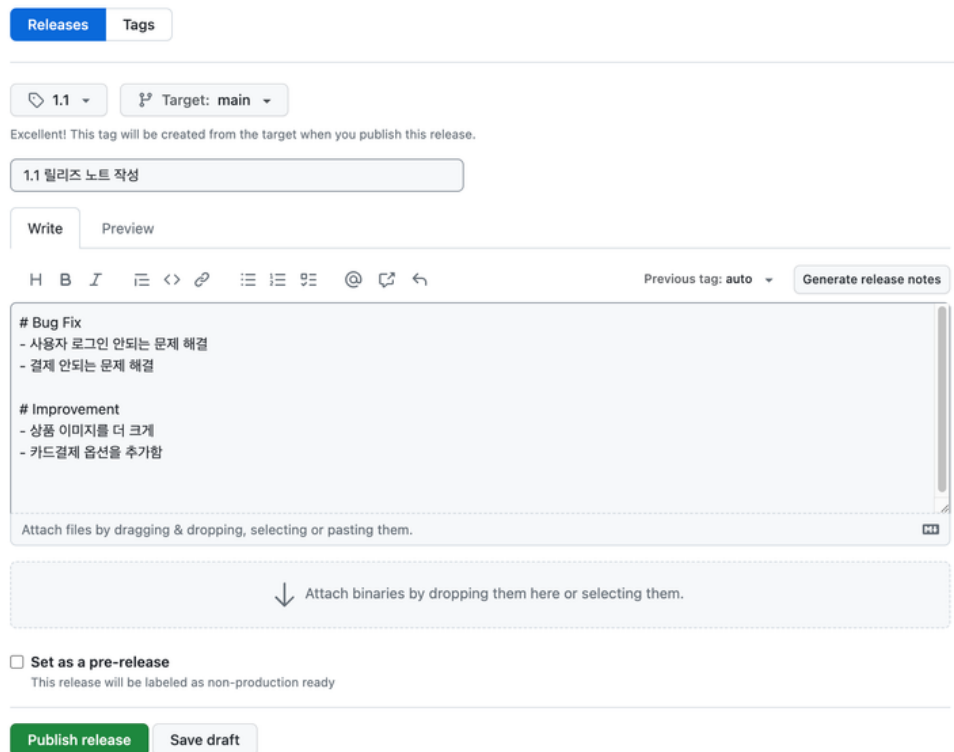
Branch를 만드는 것과 유사하게, 릴리즈할 버전을 입력하고 Create new tag를 눌러주면, tag가 생성된다.

이미 원하는 tag를 생성하면서 작업을 한 경우, tag를 선택하기만 하면 된다.



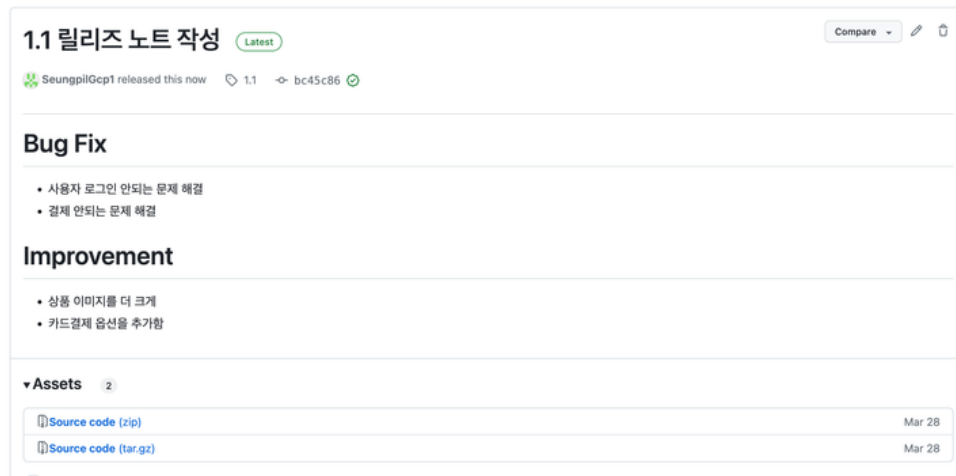
만약 새로운 태그를 생성해줬을 때는 릴리즈할 프로젝트가 담겨있는 branch를 선택해줘야 한다.

2. Release title과 Describe를 작성해주자



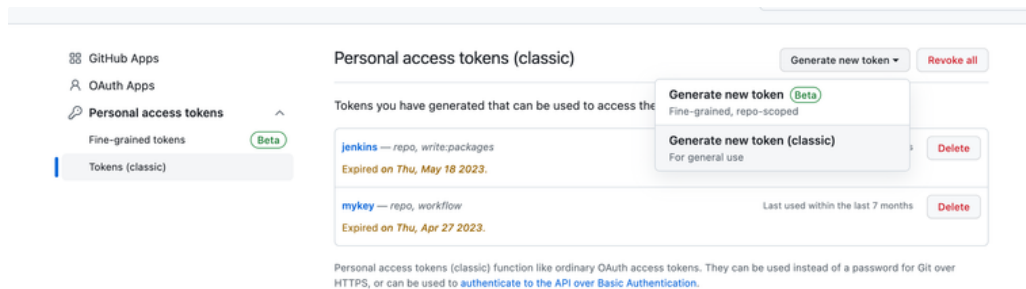
추가로, 릴리즈에 함께 빌드된 바이너리 파일을 포함하고 싶다면, 아래 공간에 Drag & Drop으로 놓아두자.

이 과정을 모두 다 마쳤다면, Publish release 버튼을 눌러주면 된다.



6. Github Package [🔗](#)

- 우측 상단 프로필 > Settings > Developer settings > Tokens → Personal access tokens → Generate new token (classic)



- 아래의 Github Package 권한으로 토큰 발급

Scope	Description	Required permission
<code>read:packages</code>	Download and install packages from GitHub Packages	read
<code>write:packages</code>	Upload and publish packages to GitHub Packages	write
<code>delete:packages</code>	Delete packages from GitHub Packages	admin
<code>repo</code>	Upload and delete packages (along with <code>write:packages</code> , or <code>delete:packages</code>)	write or admin

- 진행에 Docker 설치 필요

[🔗 Install Docker Desktop on Windows](#) 에서 알맞은 환경의 도커 데스크탑 설치

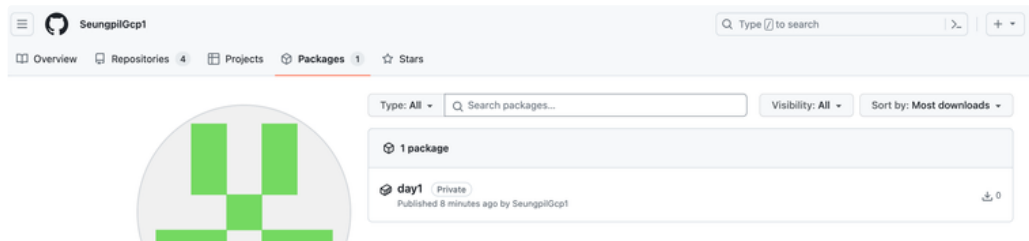
- 도커 로그인

```
1 $ docker login ghcr.io
2 Username: <username>
3 Password: <token>
4 Login Succeeded
```

- 도커 이미지 업로드 진행

```
1 # 이미지 내려받기
2 $ docker pull alpine:latest
3 latest: Pulling from library/alpine
4 579b34f0a95b: Pull complete
5 Digest: sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
6 Status: Downloaded newer image for alpine:latest
7 docker.io/library/alpine:latest
8
9 # 도커 이미지 이름을 Github 계정에 맞추어 변경
10 $ docker tag alpine:latest ghcr.io/<사용자계정_소문자로>/day1:1.0
11
12 # 도커 이미지 업로드
13 $ docker push ghcr.io/<사용자계정_소문자로>/day1:1.0
14 The push refers to repository [ghcr.io/seungpilgcp1/day1]
15 5f4d9fc4d98d: Pushed
16 1.0: digest: sha256:6ce9a9a256a3495ae60ab0059ed1c7aee5ee89450477f2223f6ea7f6296df555 size: 528
```

- <https://github.com/<사용자계정>?tab=packages> 접속
- package 에 도커 이미지 업로드 확인



- 소스코드 Repository 와 연계 설정

