

팀 프로젝트 Reopository 관리

1. 개인 github repositories 에서 repository 생성 한 후 collaborator 로 팀원 초대

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * 8253jang / Repository name *

Great repository names are short and memorable. Need inspiration? How about **special-octo-lamp** ?

Description (optional)

☒ Public
 Anyone on the internet can see this repository. You choose who can commit.

☐ Private
 You choose who can see and commit to this repository.

새로운 Repository > Settings > Collaborators > Add people

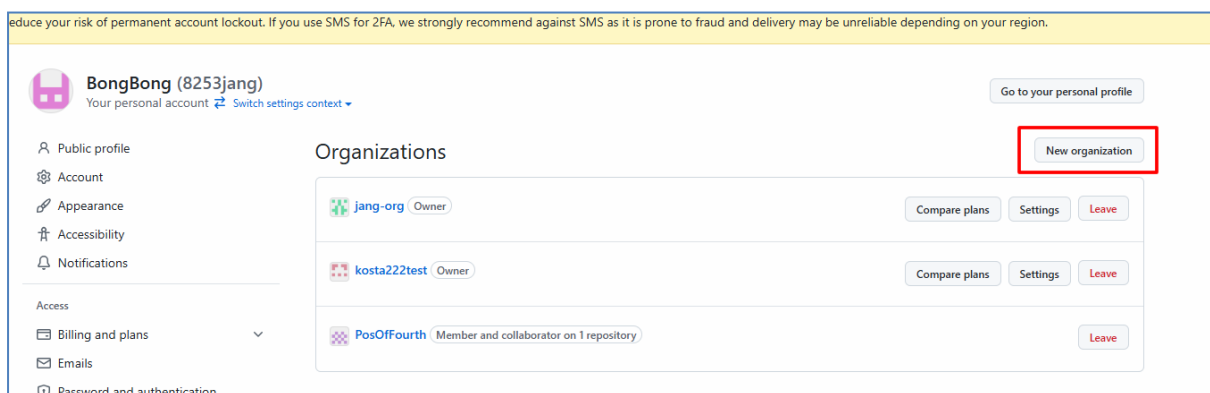
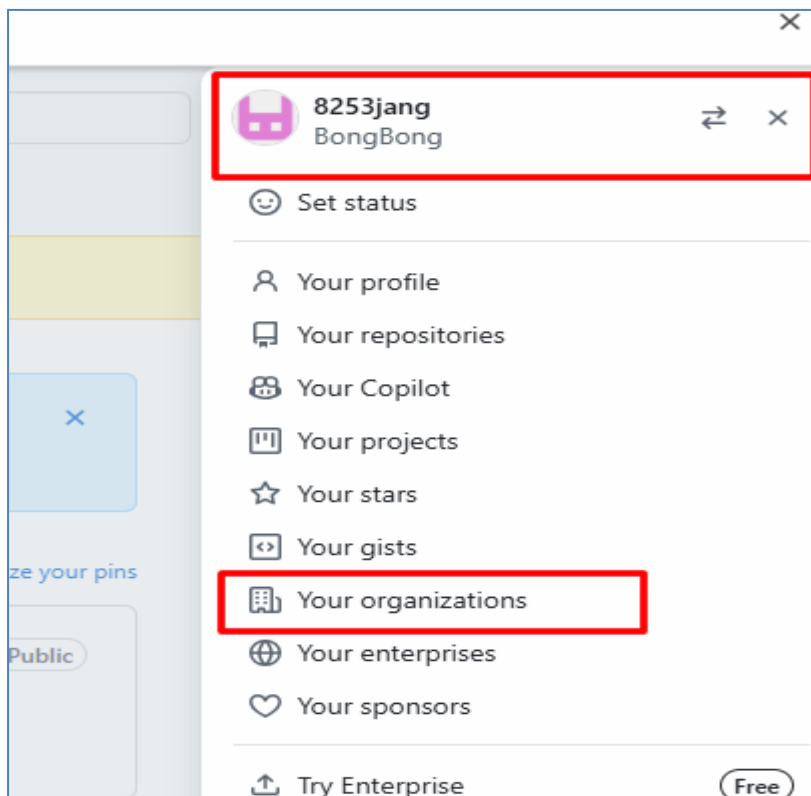
The screenshot shows the GitHub interface for a repository named 'Git_295' by user '8253jang'. The navigation path is highlighted with red boxes and numbers: 1) Repository name, 2) Settings, 3) Collaborators, and 4) Add people. A modal window titled 'Add people to Git_295' is open, showing a search bar with the text '5) 조원 이름 or email 검색' and a list of suggested users: Chanbum, chanbume, chanbum2466, and Chanbumjung, each with an 'Invite collaborator' button. The 'Add people' button at the bottom right of the modal is also highlighted with a red box and number 4).

초대된 팀원에게 초대장이 메일로 전송된다. > 팀원이 초대장을 수락하면 이제 github 을 함께 관리 할 수 있다.

2. organization 생성 후 repository 생성

- ✓ collaborator 로 팀원초대
- ✓ 권한, 팀 등 관리 상세 관리 장점
- ✓ 일반적으로 회사에서 사용하는 방식

위 1의 방식은 Repository를 생성한 팀원 github에 조원이 기여하는 방식이라면 organization은 팀원들만의 조직을 만들고 함께 관리하는 방식이다. 또한 팀원들마다 권한을 다르게 부여할 수도 있다.



Choose a plan

Pick a plan for your organization

Free
The basics for individuals and organizations
\$0 USD per user/month
[Create a free organization](#)
> Unlimited public/private repositories
> Automatic security and version updates

MOST POPULAR
Team
Advanced collaboration for individuals and organizations
\$4 USD per user/month
[Continue with Team](#)
← Everything included in Free, plus...
> Access to GitHub Codespaces

Enterprise
Security, compliance, and flexible deployment
Starting at
\$21 USD per user/month
[Start a free trial](#) [Contact Sales](#)
← Everything included in Team, plus...
> Data residency

Tell us about your organization

Set up your organization

Organization name *

→ 전세계적으로 유일해야한다.

This will be the name of your account on GitHub.
Your URL will be: <https://github.com/kosta-295>.

Contact email *

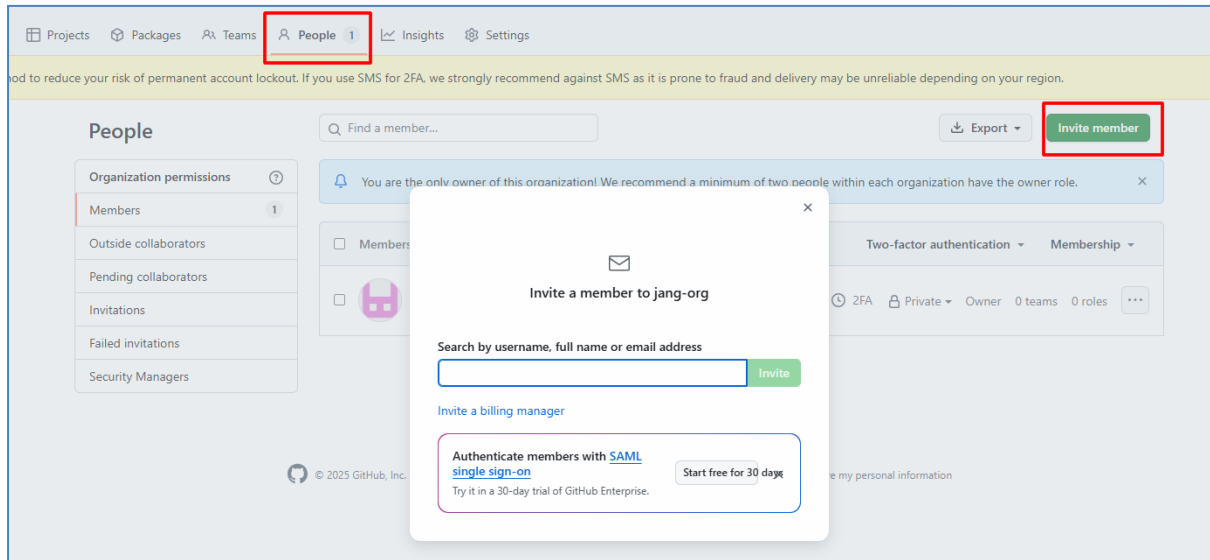
This organization belongs to:

☒ **My personal account**
I.e., 8253jang (BongBong)

☐ **A business or institution**
For example: GitHub, Inc., Example Institute, American Red Cross

Verify your account

github 주소를 확인해보자 - <https://github.com/kosta-295>



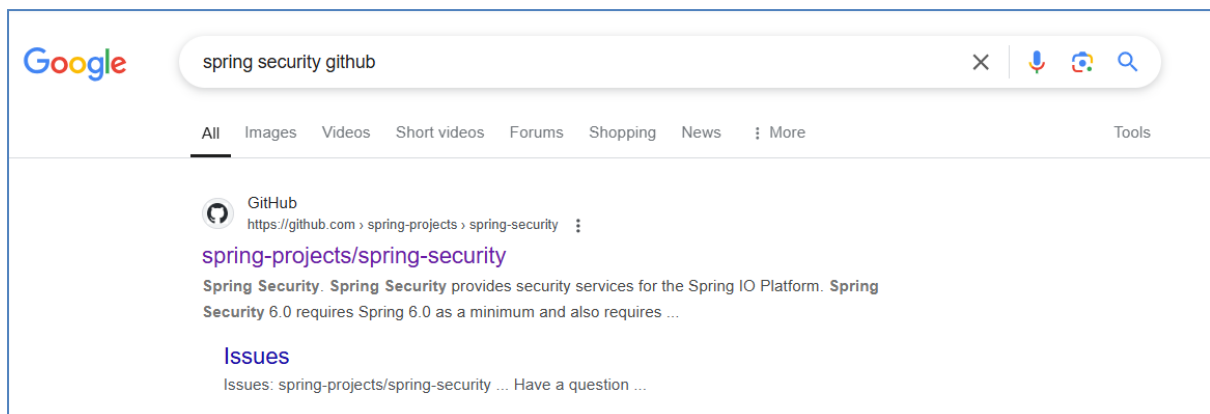
People > invite member 를 클릭해서 조원을 초대한다.

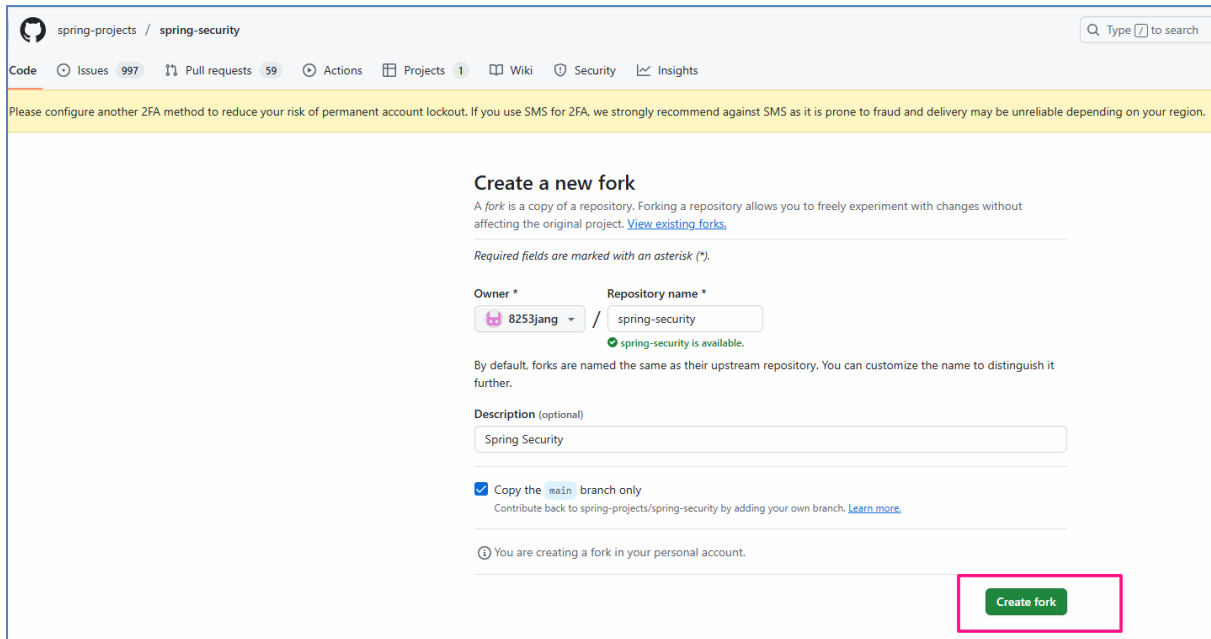
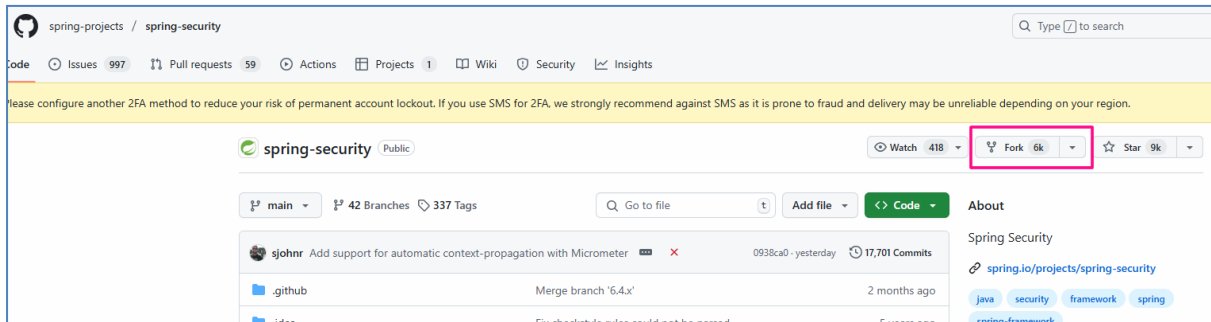
3.fork 방식

포크는 branch 를 생성해서 **push 할 수 있는 권한이 없는 외부 기여자가 Pull Request 하기 위한 목적으로 사용하는 방법**이다.

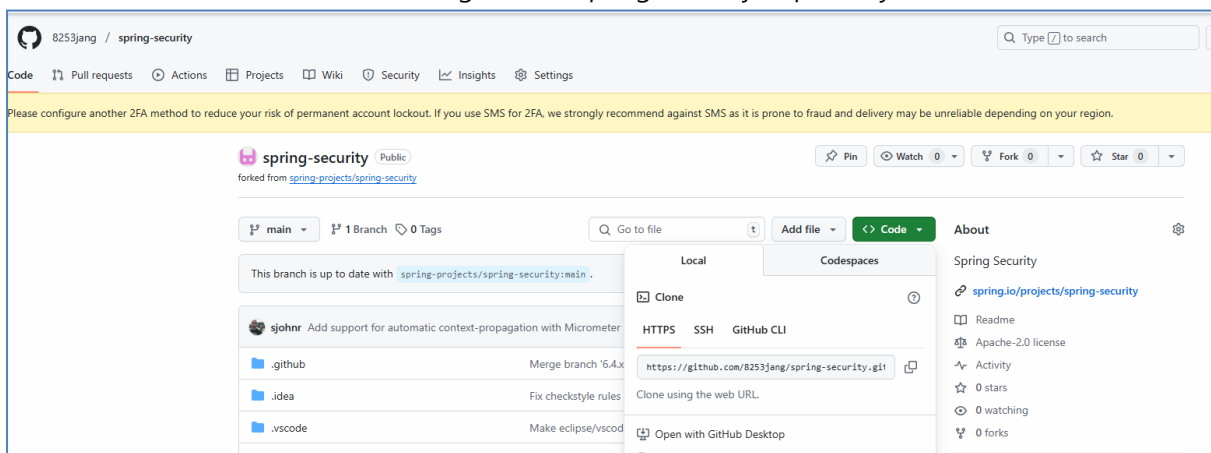
- ✓ PR 은 Repository 에 대한 권한이 없어도 요청 가능하다.
- ✓ 주로 Opensource 에 기여할 때에 일반적으로 사용하는 방식이다.

Spring security 는 오픈소스이므로 구글에서 검색을 통해 github 에 들어가 fork 해보자





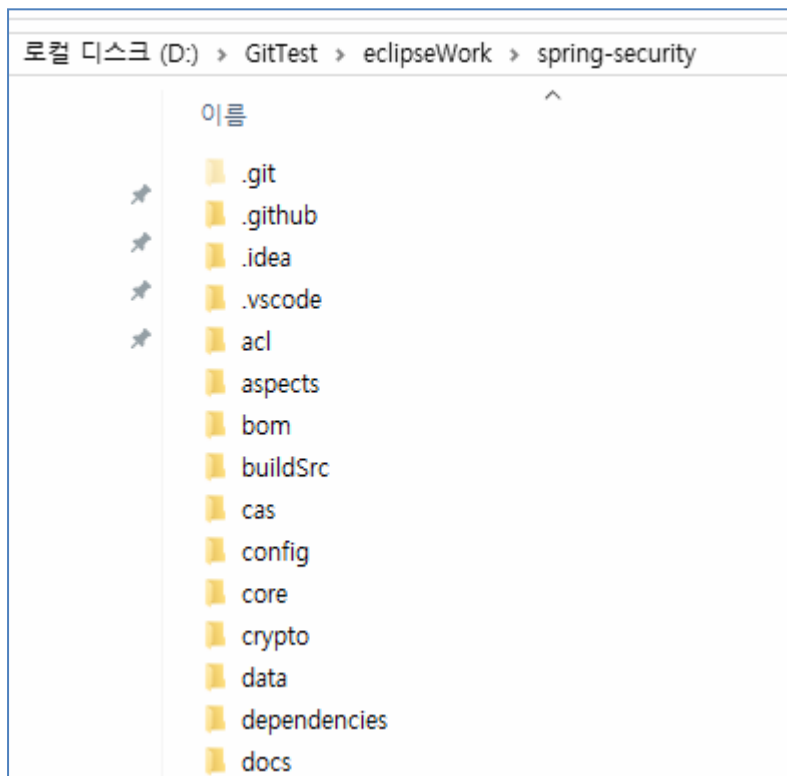
Create fork를 완료하고 나면 자신의 github에 spring-security Repository가 생긴다.



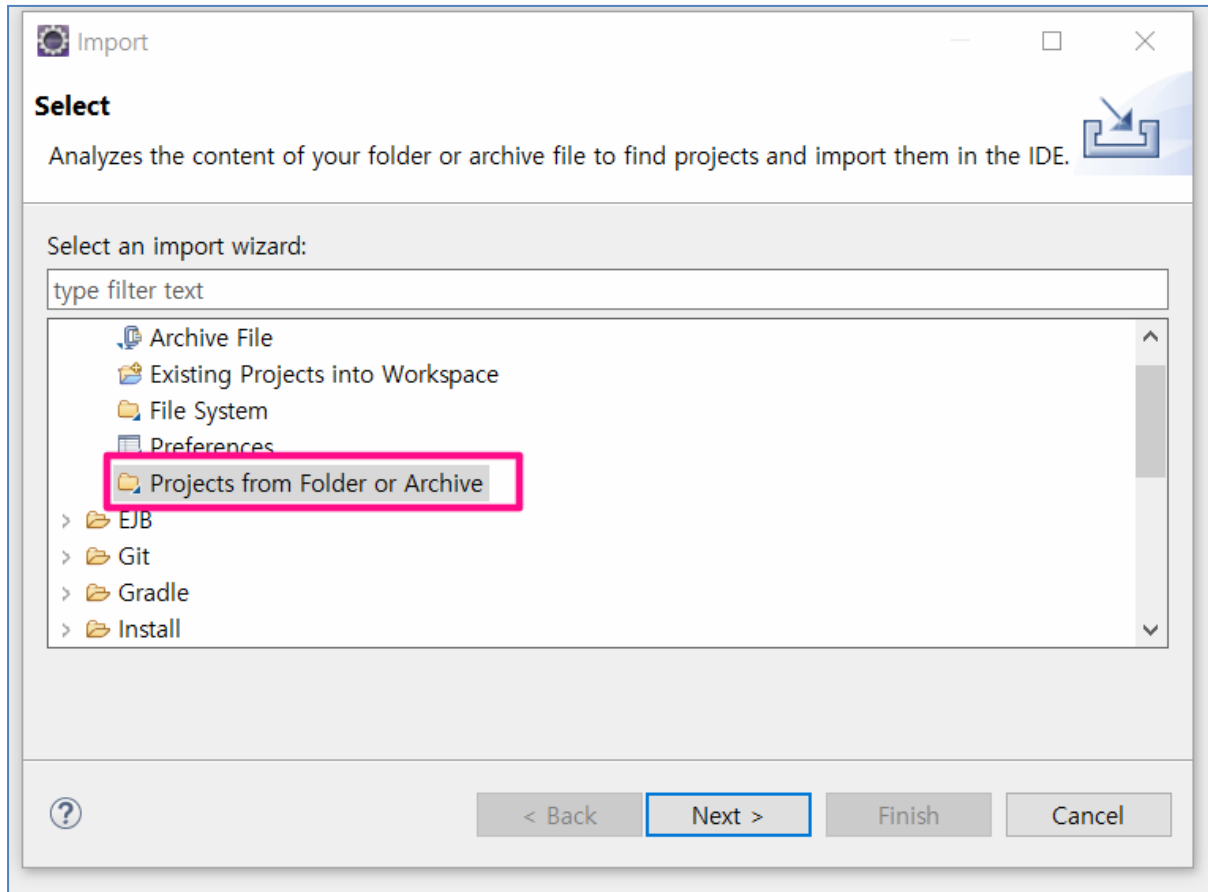
자신의 로컬 pc에 git clone한다.

```
KOSTA@DESKTOP-7P1AKSP MINGW64 /d/gitTest/eclipseWork
$ git clone https://github.com/8253jang/spring-security.git
Cloning into 'spring-security'...
remote: Enumerating objects: 346232, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 346232 (delta 9), reused 4 (delta 4), pack-reused 346199 (from 3)
Receiving objects: 100% (346232/346232), 74.40 MiB | 6.96 MiB/s, done.
Resolving deltas: 100% (172193/172193), done.
Updating files: 100% (5107/5107), done.
```

폴더에 생긴 프로젝트를 확인한다.



프로젝트를 IDE로 import 한다.



Fork한 프로젝트를 자신의 github에 push 하여 커밋 이력을 남기는 것은 얼마든지 할 수 있다.
변경된 정보를 spring-security에 기여하고 싶으면 pull Request를 요청한다.

Git tag관리

main 브랜치에서 tag 를 붙여 버전을 명시한다. 일반적으로 release(특정 버전의 소프트웨어를 배포하거나 배포 준비가 완료된 상태)를 하고자 할때 tag 사용한다.
사용자가 이용가능한 상태가 되었을 때 주로 release 한다.

관련 명령어

git tag 버전명(v1.0)

: 마지막 커밋을 대상으로 tag 설정

git tag -a v1.0 -m "message"

: tag 에 메시지 기록

git push origin 버전명(v1.0)

: 소스코드 변경사항에 대한 commit, push 와는 상관없이 별개로 진행

git tag

: 태그 목록 조회

github 에 switch 된 branch 기준으로 tag 별 release 가 생성되고,
release 에는 source 코드가 압축파일로 생성된다.

```
KOSTA@DESKTOP-7PLAKSP MINGW64 /d/GitTest/eclipsework/Git_295 (main)
$ git tag v1.0
```

```
commit d412d8c1258e18b501ae67935b168b2a077df184 (HEAD -> main, tag: v1.0, origin/main)
Merge: 4b50049 bb0465d
Author: BongBong <8253jang@daum.net>
```

```
KOSTA@DESKTOP-7PLAKSP MINGW64 /d/GitTest/eclipsework/Git_295 (main)
$ git push origin v1.0
```

Github에 저장소를 확인해본다.

