오픈소스 설계 팀 프로젝트  
과제 1

**학번** : 14011150(김병진), 15011121(손유성),   
16011126(정홍준), 18013196(주근영),

18013192(이단비)   
**팀 이름** : OSA(Open Source Architecture)

**Use case 이름** : 서버 온도 관리 시스템

**교수** : 송재승

**수업이름** : 오픈 소스 설계

목차

[오픈소스 설계 팀 프로젝트 과제 1 **오류! 책갈피가 정의되어 있지 않습니다.**](#_Toc51764768)

[1. 개요 3](#_Toc51764769)

[2. 목적 3](#_Toc51764770)

[3. Team name - OSA 3](#_Toc51764771)

[1. Members 3](#_Toc51764772)

[4. Problems to solve 3](#_Toc51764773)

[1. What’s a matter 3](#_Toc51764774)

[2. Why this should be solved. 4](#_Toc51764775)

[3. Why this causes an issue to campus or normal our life 4](#_Toc51764776)

[4. Who is the target customer for this service. 4](#_Toc51764777)

[5. Use case for your project 4](#_Toc51764778)

[6. High-level architecture 5](#_Toc51764779)

[1. Which components are composing the architecture 5](#_Toc51764780)

[2. Why these components are required.. description of each component 5](#_Toc51764781)

[3. How these components can deliver a required service for the project 6](#_Toc51764782)

[7. How people who use your smart service can get benefit from the service you are proposing 6](#_Toc51764783)

[8. Project status 6](#_Toc51764784)

# 개요

이 레포트는 오픈 소스 설계 팀 프로젝트를 위해서 작성되었다. Open source를 이용해서 어떤 IoT service를 구현하고 data를 어떻게 주고 받게 할 것인지를 정한다.

# 목적

Open source를 이용해서 IOT 설계의 방향성을 잡는다.

# Team name - OSA

## Members

|  |  |
| --- | --- |
| 김병진 | 라즈베리 파이를 이용하여 서버실의 온도를 측정하고, 온도에 따라 작동할 팬 모터의 작동을 위한 Application Entity를 구현합니다. |
| 손유성 | 사용자에게 서버실의 온도를 실시간으로 보여줄 수 있는 웹 페이지 및 앱 서비스를 구현합니다. |
| 주근영 | 사용자가 웹 화면을 통하여 실시간으로 온도를 확인할 수 있는 웹 페이지를 구현합니다. |
| 이단비 | 인프라를 통해 데이터의 명령을 주고받을 Common Service Entity를 구현합니다. |
| 정홍준 | 인프라를 구축합니다. 인프라를 통해 데이터 명령을 주고받을 Common Service Entity를 구현하여 플랫폼을 구축합니다. |

# Problems to solve

## What’s a matter

서버실의 장비들은 특정온도가 유지되지 않는다면 여러 문제점을 가져올 수 있다. 온도가 너무 높을 경우 shutdown의 위험성은 물론이고, 고가의 장비가 고장을 일으킬 수 있으며 온도가 너무 낮을 경우 특정장비들은 제 기능을 100% 발휘하지 못하기 때문에 항상 18 - 24도의 온도를 유지하는 것이 기본이다. 하지만 이를 수동으로 조절하기에는 매우 번거로우며 원격으로 온도를 조절할 수 없다면 긴급한 상황에서 즉각적인 대처를 못할 것이고, 이는 큰 사고로 이어질 수 있다.

## Why this should be solved.

서버실의 온도가 기준치를 벗어났을 경우 자동으로 온도를 조절해주고, 사용자에게 알람을 보내는 시스템을 구축한다면 매번 수동적으로 서버실의 온도를 확인하고, 조절하는 필요성은 사라질 것입니다. 나아가 원격으로 현재 서버실의 온도를 확인하고, 조절할 수 있다면 사용자가 서버실에서 멀리 떨어져 있을 경우에도 쉽게 서버실의 온도를 통제할 수 있을 것입니다.

## Why this causes an issue to campus or normal our life

우리 세종대학교에도 서버실이 있으며 만약 이 서버실에 특정한 문제가 생겨 온도가 급격하게 올라가 shutdown이 되어버린다면 학생들이 수강신청을 하거나 학교 홈페이지에서 지식을 습득할 때 큰 문제가 될 것입니다. 또한, 데이터 센터나 AI센터에서 보관하고 있던 데이터나 작업에도 문제가 생길 수가 있습니다.

## Who is the target customer for this service.

서버를 운영하거나 관리하는 담당자.

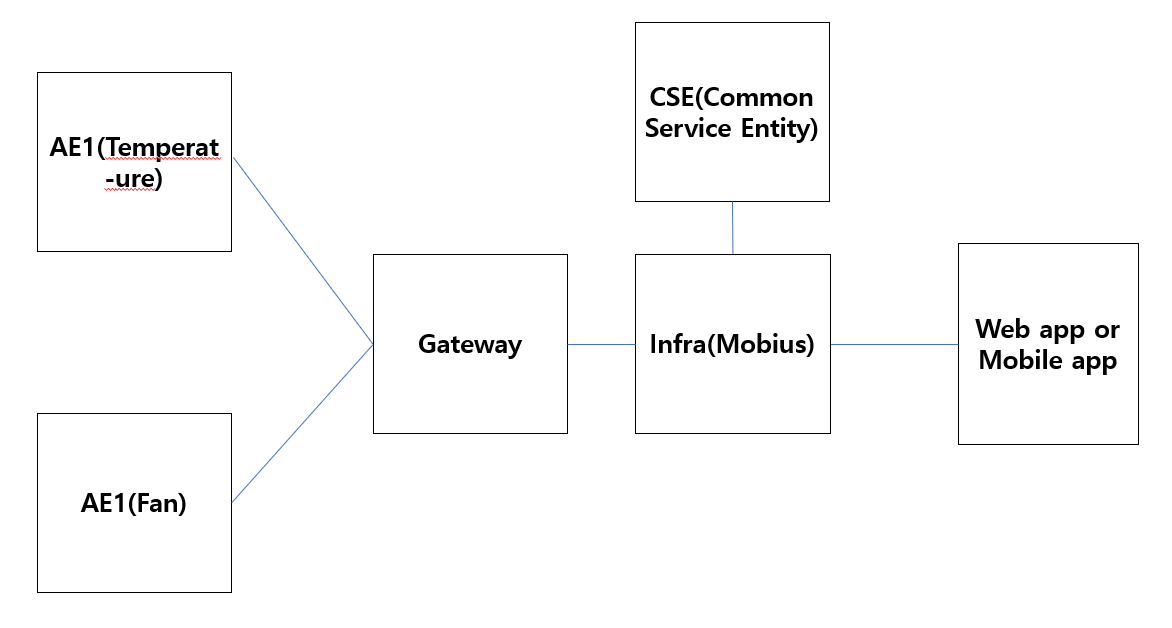
# Use case for your project

프로젝트 이름은 서버 온도 관리 시스템이다. 냉각기의 역할을 하는 Actuator로는 모터를 사용할 것이다.

* Assumptions / Pre-conditions => 온도가 상한치를 넘어가면 냉각 시스템을 가동하고 관리자에게 알린다.
* Post-conditions => Main 냉각기의 문제를 해결하면 관리자는 web app을 통해서 예비 냉각기를 정지 시킬 수 있다. 하지만 현재 온도가 적정 온도보다 높을 경우, 관리자도 예비 냉각기의 가동을 중지시킬 수 없다. 적정 온도가 되었을 때 냉각 시스템을 중지 시킬 수 있다.
* Business rules => 26C가 될 때 냉각시스템을 가동해서 20C로 맞춘다. 30C가 되었을 경우 모터 회전 속도를 급속 증가 시킨다.
* Process flow => AE1에서 온도를 측정해서 mobius에 데이터를 전달한다. Web application은 mobius를 통해서 실시간으로 온도를 체크한다. CSE(Common Service Entity)는 mobius의 온도를 확인하면서 30도가 되었을 때 AE1에게 냉각 시스템을 가동하라고 명령한다. 동시에 관리자에게 해당 room에 문제가 있다고 알린다.

# High-level architecture

## Which components are composing the architecture



## Why these components are required.. description of each component

* Temperature (ADN-AE1) : 온도를 측정하기 위한 온도 센서다.
* Fan (ADN-AE2) : 온도의 변화를 위한 팬 모터다.
* Gateway (MN-CSE) : 여러 기기들이 연결되어 서비스 플랫폼에 데이터 전달 등을 담당한다.
* Open M2M platform (IN-CSE) : 디바이스들을 리소스로 등록하고 센서와 통신한다.
* Application (IN-AE) : 스마트폰 디바이스에 내장된 어플리케이션으로서 oneM2M 서비스 플랫폼인 IN-CSE와 직접 상호작용하여 팬모터를 원격으로 제어하는 기능을 가지고 있다.

## How these components can deliver a required service for the project

* 온도 센서는 온도 데이터를 측정하여 gateway를 통해 openM2M 플랫폼으로 전송한다.
* oneM2M 플랫폼은 검색 기능과 구독 기능을 이용하여 원하는 데이터를 어플리케이션과 MQTT로 통신한다.
* 어플리케이션은 스마트폰 디바이스에 내장되어 oneM2M 서비스 플랫폼과 직접 통신하여 사용자에게 서비스를 제공한다.
* 어플리케이션 혹은 자동화에 의해 팬모터가 작동된다.

# How people who use your smart service can get benefit from the service you are proposing

관리자가 서버실에 24시간 지키고 있지 않아도 안심하고 다른 작업을 할 수가 있다. 관리자가 서버로부터 물리적으로 멀리 떨어져 있고 예기치 못 하게 냉각 시스템이 작동하지 않더라도 안심할 수가 있다. 또한, 실시간으로 언제 어디서든지 서버실의 상황을 체크할 수가 있다.

# Project status

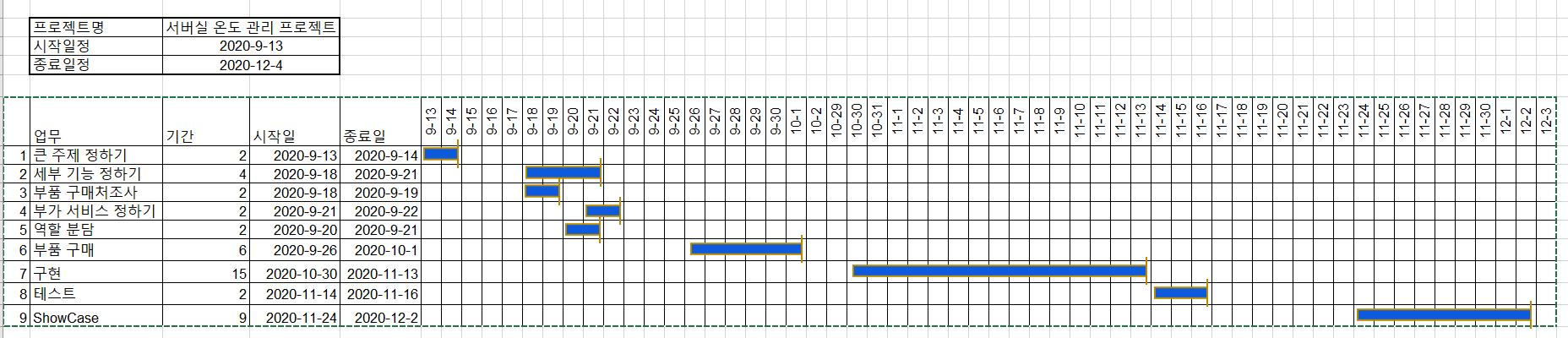


그림 1 Project status - Gantt chart

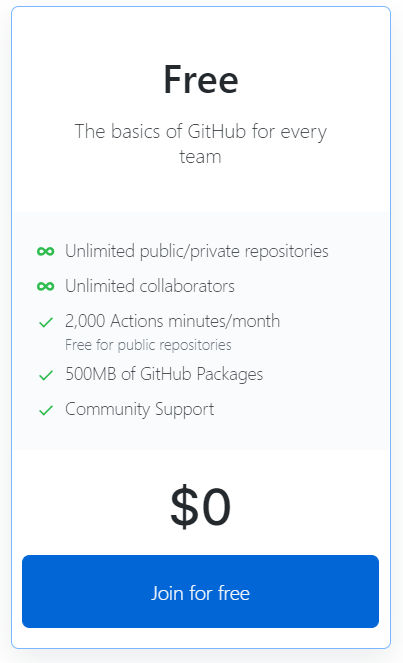
# Git

## Organization

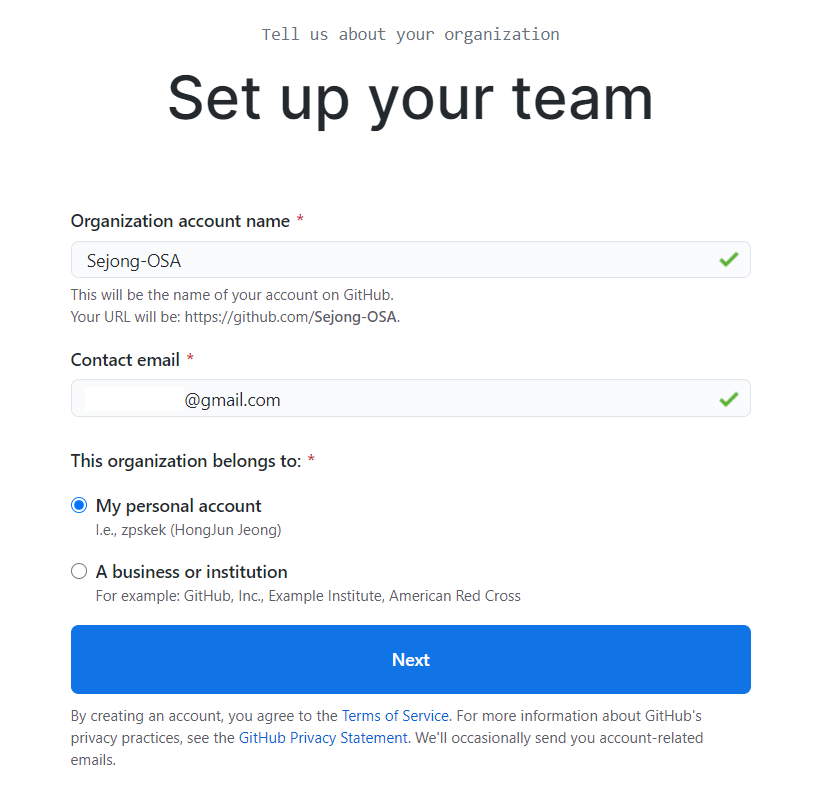
1. New oraganization



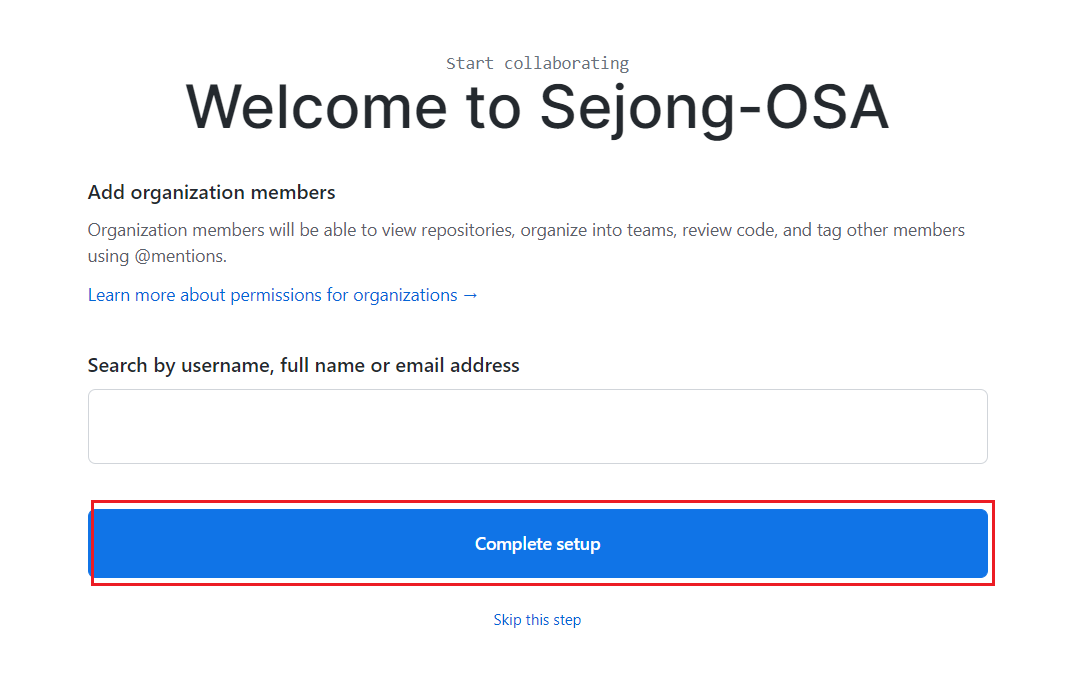
1. Select free



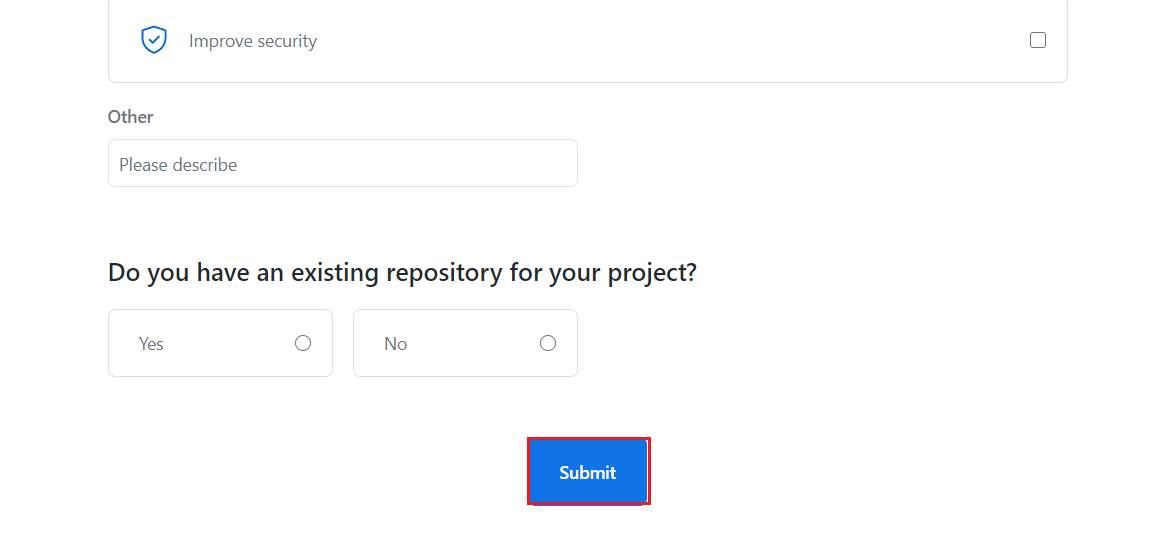
1. Set up our team



1. Complete setup

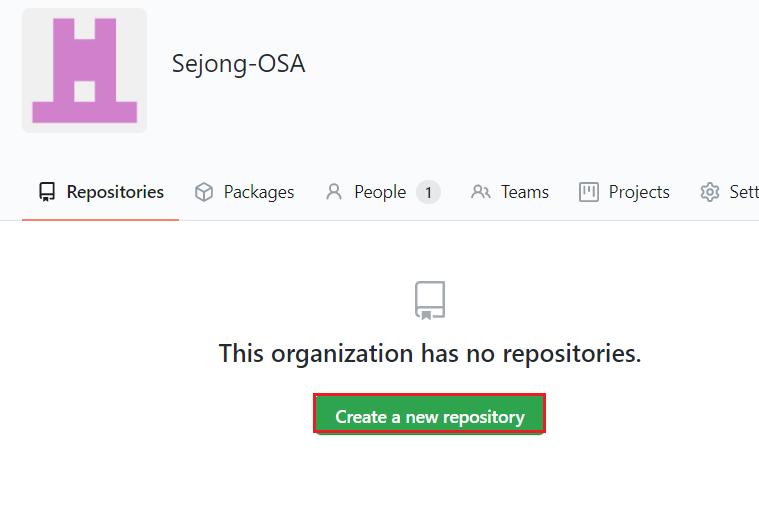


1. It doesn’t matter to check the survey or not.

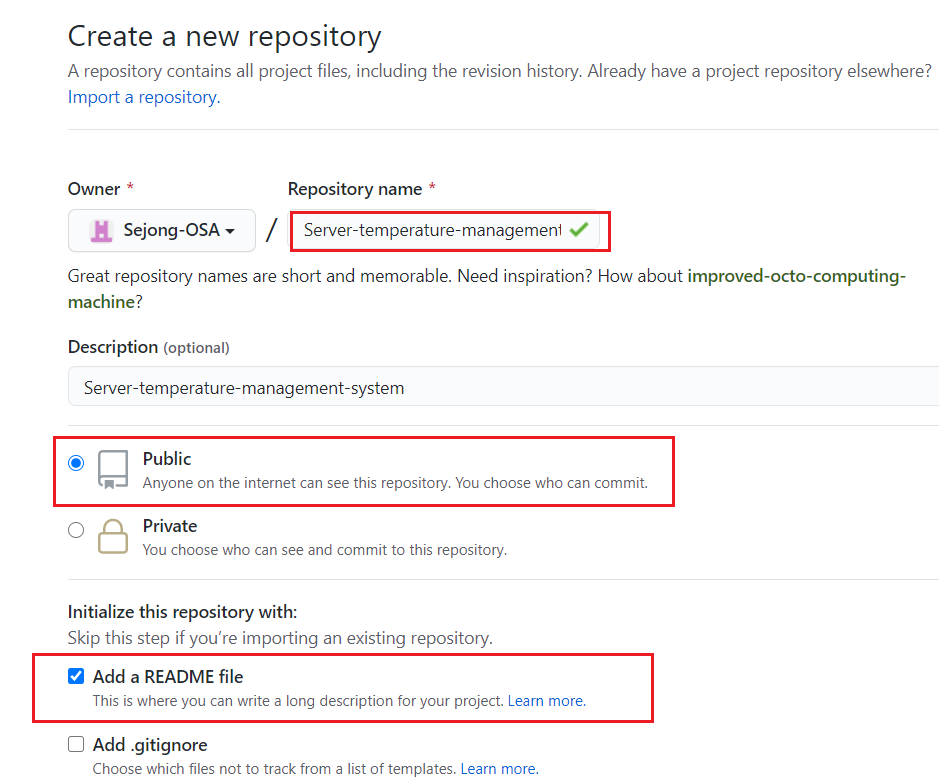


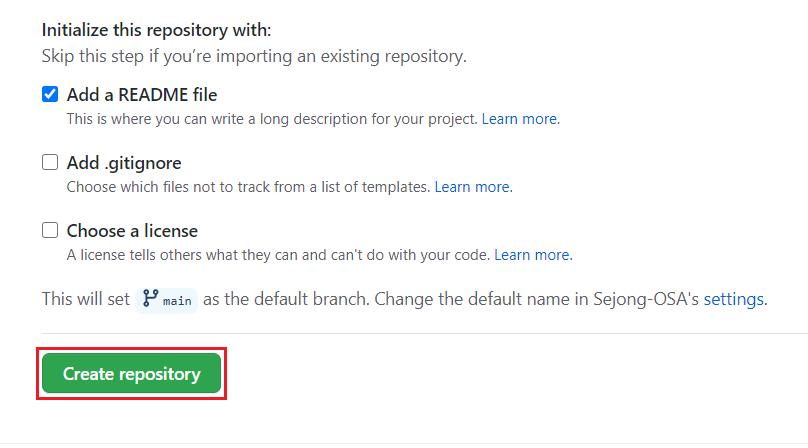
## Create a new repository

1. Create a new repository 클릭



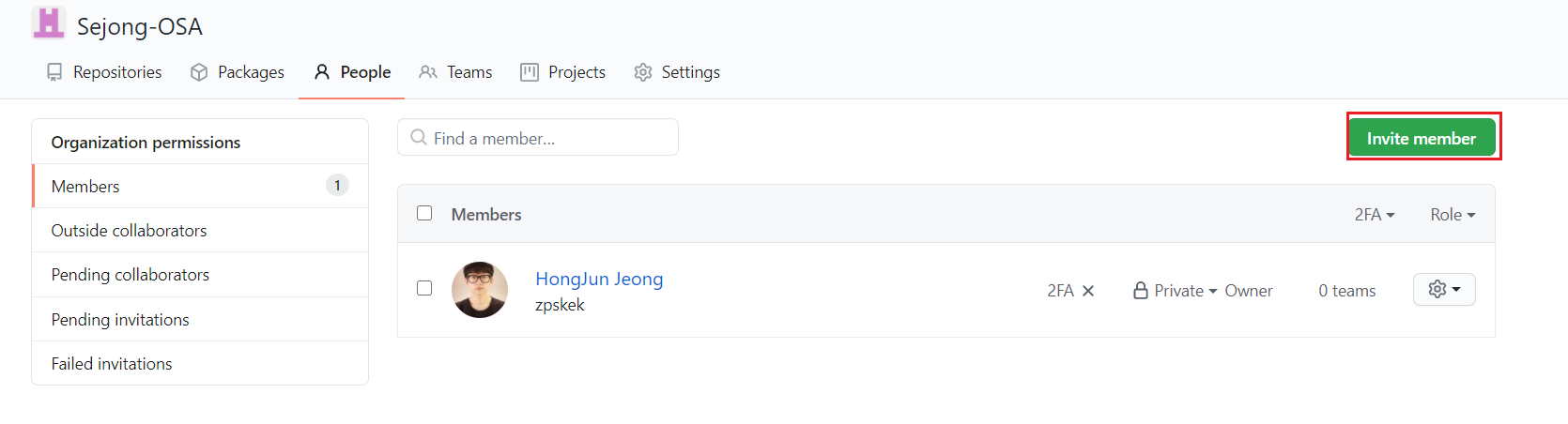
1. 빈칸 작성



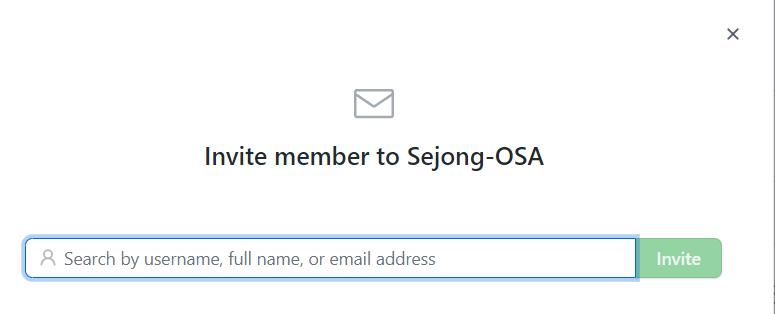


## Invite member

1. Invite member 클릭

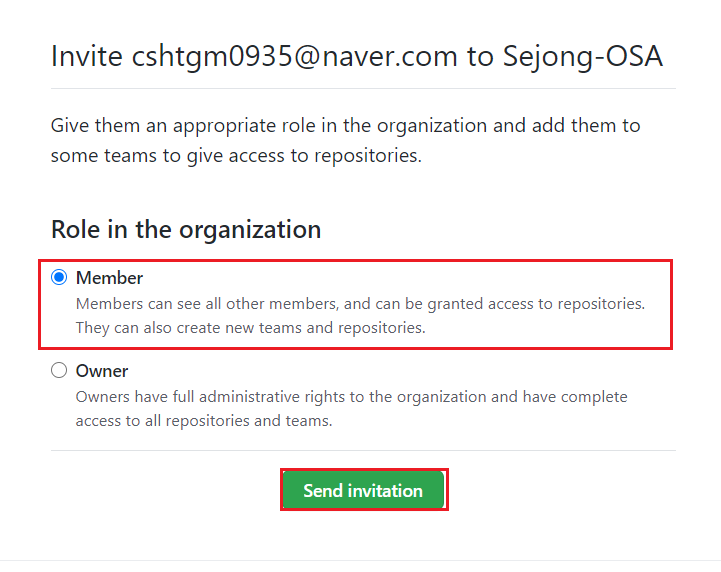


1. 팀원의 아이디를 입력하고 초대한다.



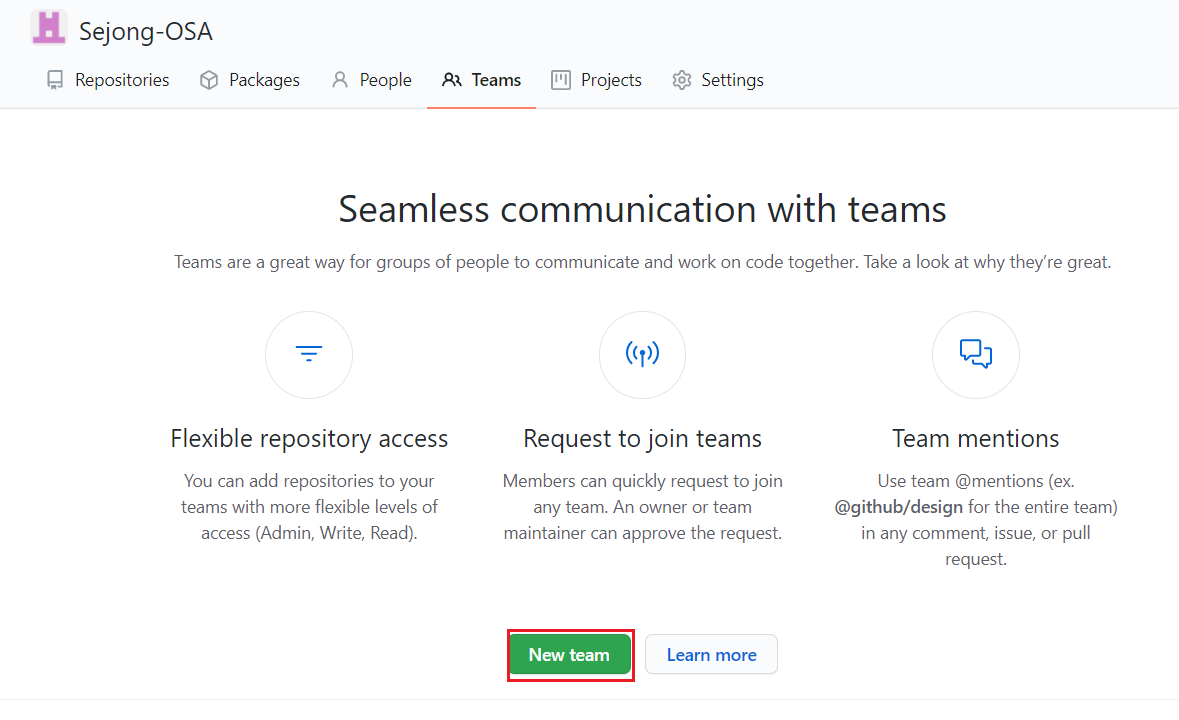
* 1. [cshtgm0935@naver.com](mailto:cshtgm0935@naver.com)
  2. [danbi9612@gmail.com](mailto:danbi9612@gmail.com)
  3. [spg9629@gmail.com](mailto:spg9629@gmail.com)
  4. [kbj2060@naver.com](mailto:kbj2060@naver.com)

1. 팀원을 Member나 Owner 중 하나를 택하고 초대한다.

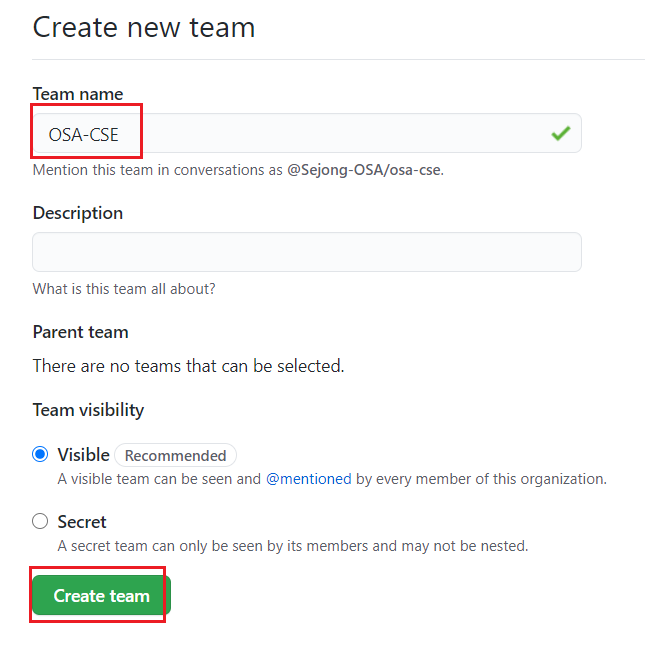


## Create a new team

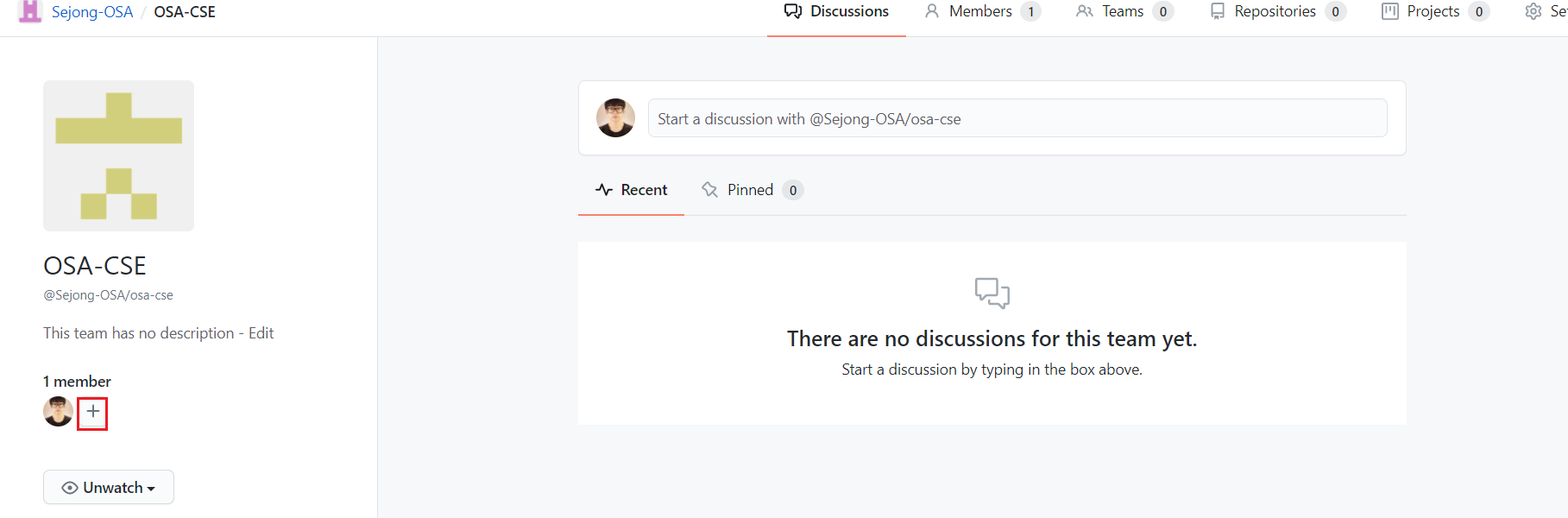
1. New team 클릭



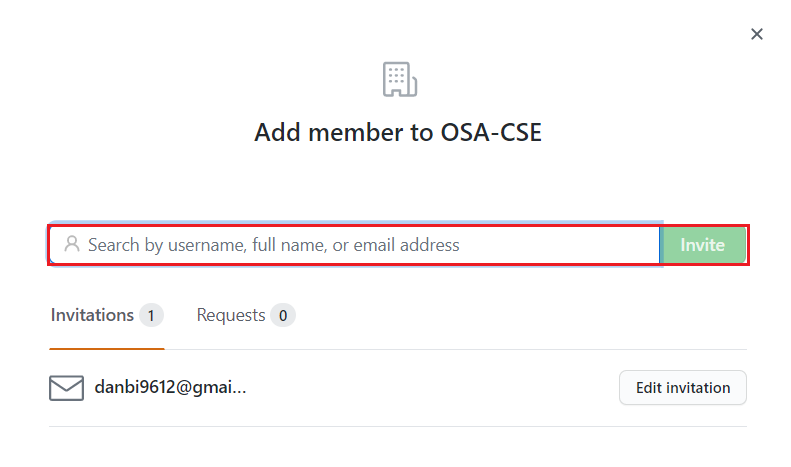
1. 팀 이름을 만들고 **Create team** 클릭ㄹ



1. 멤버 추가



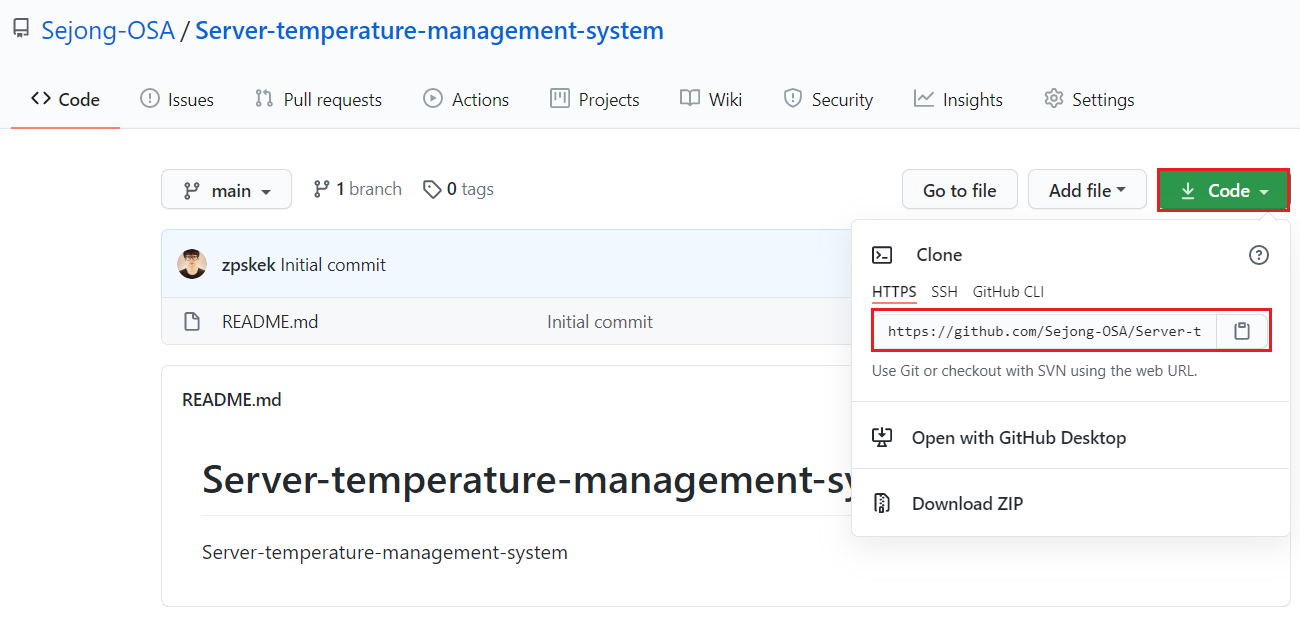
1. 입력란에 팀원 이름을 추가하고 Invite한다.



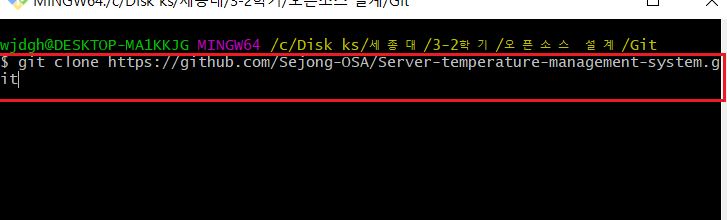
## Git

### Git 접속 방법

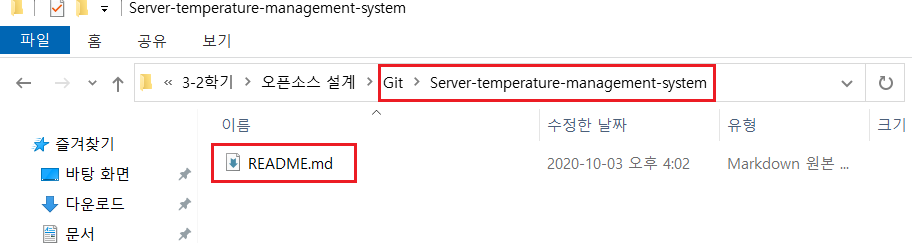
1. git clone



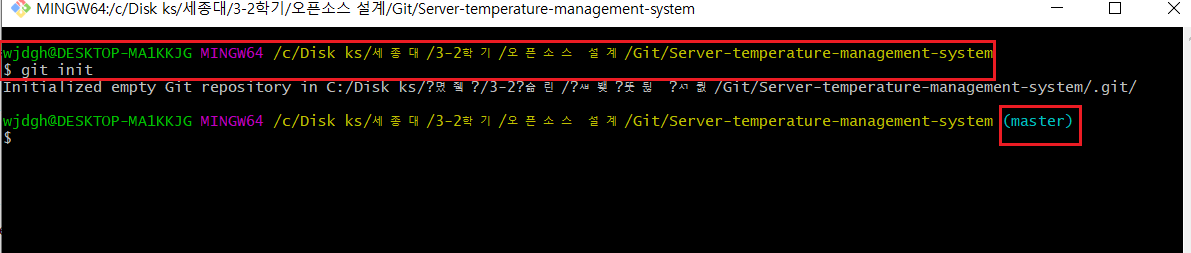
* 1. Code 버튼을 누르고 해당 주소를 복사한다.



* 1. $ git clone <복사한 주소>
     1. 해당 명령어를 통해서 원격 repository에 있는 source를 가져와서 local에 저장한다.



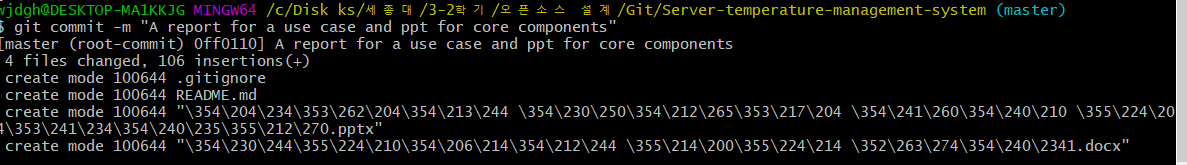
1. git init



1. **git init**을 통해 **.git**이라는 하위 디렉토리를 생성한다. 이를 통해서 **git add**와 **git commit**으로 파일들을 커밋할 수 있다.
2. git add



1. git add를 통해서 .git이 있는 폴더의 모든 파일(및 폴더)를 staging area에 올릴 수 있다.
2. git commit



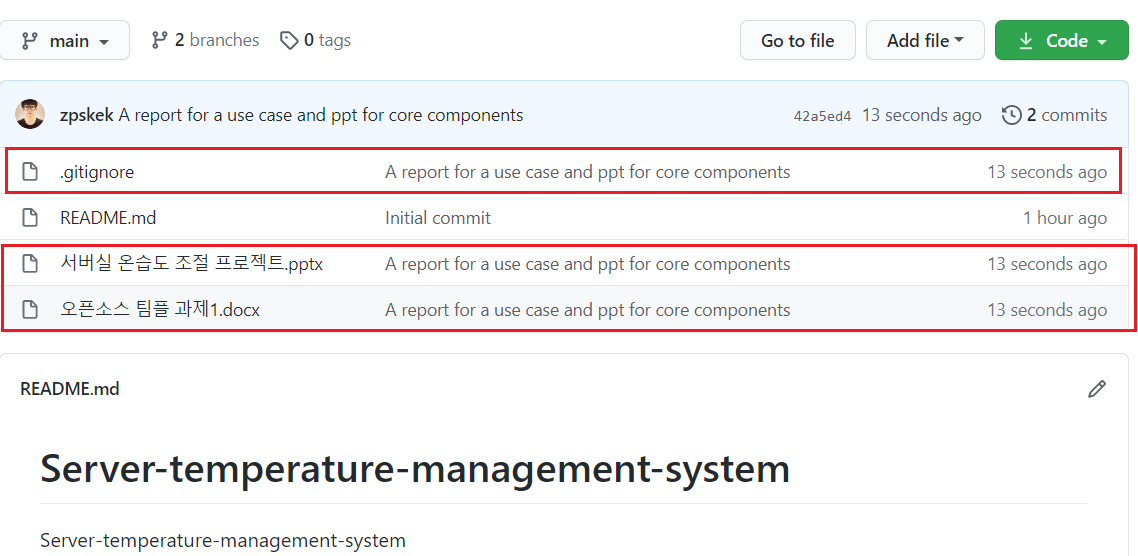
1. -m 옵션을 사용해서 메시지와 함께 commit 할 수 있다.
2. git push
   1. git remote add origin <address>



* + 1. 위 명령어를 통해서 원격 저장소의 주소를 설정한다.
  1. git push -u origin master



* + 1. git push 명령어를 통해서 원격 저장소에 파일들을 저장할 수 있다. 이 때 origin은 원격 저장소의 주소이고 master는 branch를 뜻한다.
    2. -u 옵션은 다음에 push할 때 origin master를 생략할 수 있게 해준다.
  1. 성공적으로 remote repository에 push했다.



### Git 주소

[**https://github.com/Sejong-OSA/Server-temperature-management-system/tree/main**](https://github.com/Sejong-OSA/Server-temperature-management-system/tree/main)