

РОЗДІЛ 7. УПРАВЛІННЯ ІТ ПРОЄКТОМ В GITHUB.

7.1. Управління ІТ-проєктом в GitHub

Типова загальна послідовність ведення проєкту ІТ в GitHub:

1. Реєстрація особистого акаунту GitHub.
2. Створення організації.
3. Створення команд, що належать до організації та запрошення учасників як в команду так і вільних учасників проєкту, що не належать до команд. Призначення ролей учасникам.
4. Поділ проєкту на функції та розподіл їх виконання між командами та учасниками проєкту. Вибір моделі контролю версій ведення проєкту – магістральний метод чи Git-flow.
5. Відповідно до вимог ІТ проєкту в організації створюються репозиторії. В GitHub, в залежності від специфіки проєкту, репозиторії створюють, як окремі проєкти, або як окремі функції проєкту.
6. До співпраці над окремими частинами проєкту (окремі репозиторії) також можливо створювати команди та запрошувати окремих вільних працівників (поза командою). Призначення ролей учасників.
7. Робота з гілками проєкту.

Створюють основні гілки **main** та **develop**. Гілку **main** використовують як гілку в яку подаються лише закінчені релізні версії програмного продукту. В процесі роботи над проєктом до гілки розробки (**develop**) додають (**комміти**) гілки окремих функцій проєкту. По завершенню робіт над функціями чи проєктом у процесі доведення до релізного варіанту проєкту виникають гілки виправлення, що враховують зміни до початкового варіанту релізу програмного проєкту чи функцій. Узгоджений, перевірений та протестований варіант з гілки **develop commit**-ом подають до релізної гілки **main**.

8. Складні ситуації злиття різних гілок внаслідок праці над ними різних команд та учасників проєкту вирішуються злиттям за узгодженням та під контролем учасників проєкту покрово.

7.2. Початкові налаштування Git на локальному пристрої

До складу **Git** входить утиліта **git config**, яка дозволяє переглядати та налаштовувати параметри, що контролюють усі аспекти роботи **Git**, а також його зовнішній вигляд. Ці параметри можуть бути збережені у трьох місцях [15]:

1. Файл *[path]/etc/gitconfig* містить значення, загальні для всіх користувачів системи та всіх їх репозиторіїв. Якщо при запуску *git config* вказати параметр *--system*, параметри читаються і зберігаються саме в цей файл. Так як цей файл є системним, то вам потрібні права супер-користувача для внесення змін до нього.

2. Файл *~/.gitconfig* або *~/.config/git/config* зберігає установки конкретного користувача. Цей файл використовується при вказівці параметра *--global* і застосовується до всіх репозиторіїв, з якими ви працюєте у поточній системі.

3. Файл *config* у каталозі **Git** (тобто *.git/config*) репозиторію, який ви використовуєте в даний момент, зберігає налаштування конкретного репозиторію. Ви можете змусити **Git** читати і писати в цей файл за допомогою *--local*, але насправді це значення за замовчуванням. Не дивно, що вам потрібно бути десь у репозиторії **Git**, щоб ця опція працювала правильно.

Налаштування на кожному наступному рівні підміняють налаштування з попередніх рівнів, тобто значення *.git/config* перекривають відповідні значення *[path]/etc/gitconfig*.

У системах сімейства **Windows Git** шукає файл *.gitconfig* у каталозі **\$HOME** (**C:\Users\USER** для більшості користувачів). Крім того, **Git** шукає файл *[path]/etc/gitconfig*, але вже щодо кореневого каталогу **MSys**, який знаходиться там, куди ви вирішили встановити **Git** під час запуску інсталятора.

Якщо ви використовуєте **Git** для **Windows версії 2.x або новіше**, то також обробляється файл конфігурації рівня системи, який має шлях **C:\Documents and Settings\All Users\Application Data\Git\config** у **Windows XP** або **C:\ProgramData\Git\config** у **Windows Vista** та новіший. Цей файл може бути змінений лише за допомогою команди *git config -f <file>*, запущеної з правами адміністратора.

Щоб переглянути всі встановлені налаштування та дізнатися, де саме вони задані, використовуйте команду:

```
$ git config --list --show-origin
```

Дані користувача:

Для того, щоб приступити до роботи необхідно ідентифікувати користувача (опція *-global* застосовує налаштування для всіх операцій в системі):

```
$ git config -global user.name «John Doe»
```

```
$ git config -global user.email «johndoe@example.com»
```

Вибір редактору:

```
$ git config --global core.editor emacs
```

Налаштування гілки за замовчуванням:

```
$ git config --global init.defaultBranch main
```

Перевірка налаштувань:

```
git config --list
```

7.3. Організація. команда. репозиторій. ролі

Організаційною основою роботи з **GitHub** є “Організації” в просторі яких знаходяться їх проекти. Облікові записи організацій представляють групи людей з сумісним правом власності проектів та різними інструментами для керування групами.

Щоб створити організацію необхідно обрати іконку “+” нагорі праворуч на будь-якій сторінці **GitHub** та виберіть у меню “**New organization**” (нова організація) (рис. 17.1).

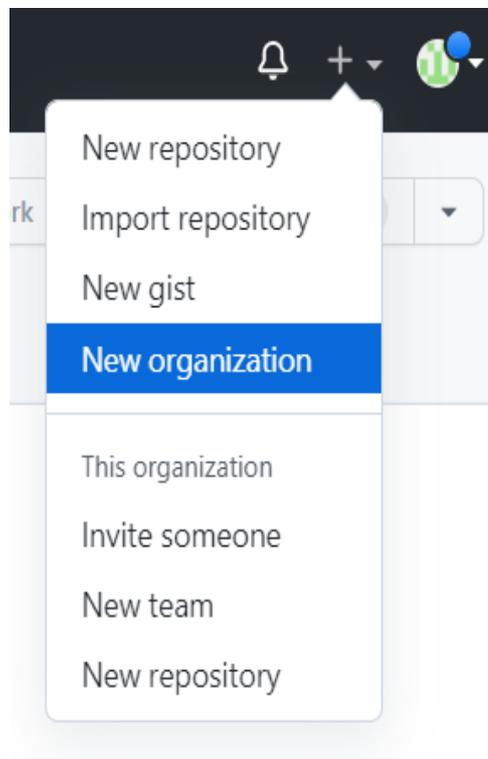


Рис. 17.1 Створення нової організації

Переходимо до сторінки вибору (рис. 17.2).

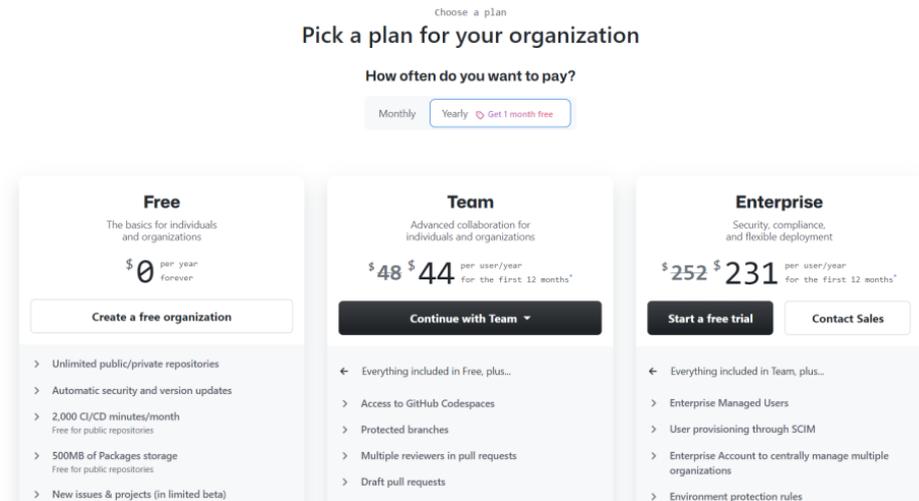


Рис. 17.2 Вибір плану для вашої організації

Де обираємо план “Free” і потім обираємо назву організації та контактну адресу електронної пошти для організації (рис. 17.3).

Tell us about your organization

Set up your organization

Organization account name *

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

Contact email *

This organization belongs to: *

My personal account
I.e., GitTEF-APEPS

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

Verify your account

I hereby accept the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

Рис. 17.3 Сторінки початкових налаштувань організації

Далі обираєте варіант вашого аккаунта – чи персональний, чи бізнес організації. Проводите верифікацію та підтверджуєте згоду і обираєте продовжити (Next).

Наступною сторінкою буде сторінка завершення створення організації, що пропонує пошуком додати учасників до організації шляхом пошуку ім'я користувача, повного ім'я або адреси електронної пошти (рис. 17.4).

The screenshot shows a page titled "Welcome to Test-APEPS2022" with the sub-header "Start collaborating". Below the title is a section "Add organization members" with a description: "Organization members will be able to view repositories, organize into teams, review code, and tag other members using @mentions." and a link "Learn more about permissions for organizations →". There is a search input field with the placeholder text "Search by username, full name or email address". At the bottom of the form is a green button labeled "Complete setup".

Рис. 17.4 Сторінка додання членів до організації

Наступна сторінка є опитувальною (рис. 17.5).

The screenshot shows a "Welcome to GitHub" survey page. It starts with the text "Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there." The survey is titled "Tell us about you" and asks "What do you spend time on most day-to-day?". Below this is a sub-question "Please select all that apply" and four checkboxes with icons: "Writing code", "Managing and coordinating engineering work", "Planning projects", and "Billing administration". There is also an "Other" section with a text input field "Please describe". The second part of the survey is titled "Tell us about your team" and asks "How many people do you expect to actively work within this GitHub organization?". It has five radio button options: "0", "1-5", "6-15", "16-24", and "25+".

Рис. 17.5 Сторінка “Welcome to GitHub”

Використовуючи в правому верхньому куті значок вашого профілю ви можете переглянути інформацію за вашою організацією (рис. 17.6). Де ви отримаєте доступ до різних даних вашої організації.

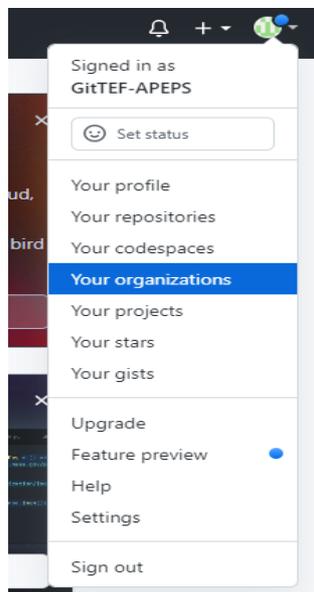


Рис. 17.6 Перехід до сторінки інформації про організацію

Також доступ до можливості управляти організаціями розташовано в лівому верхньому куті (рис. 17.7).

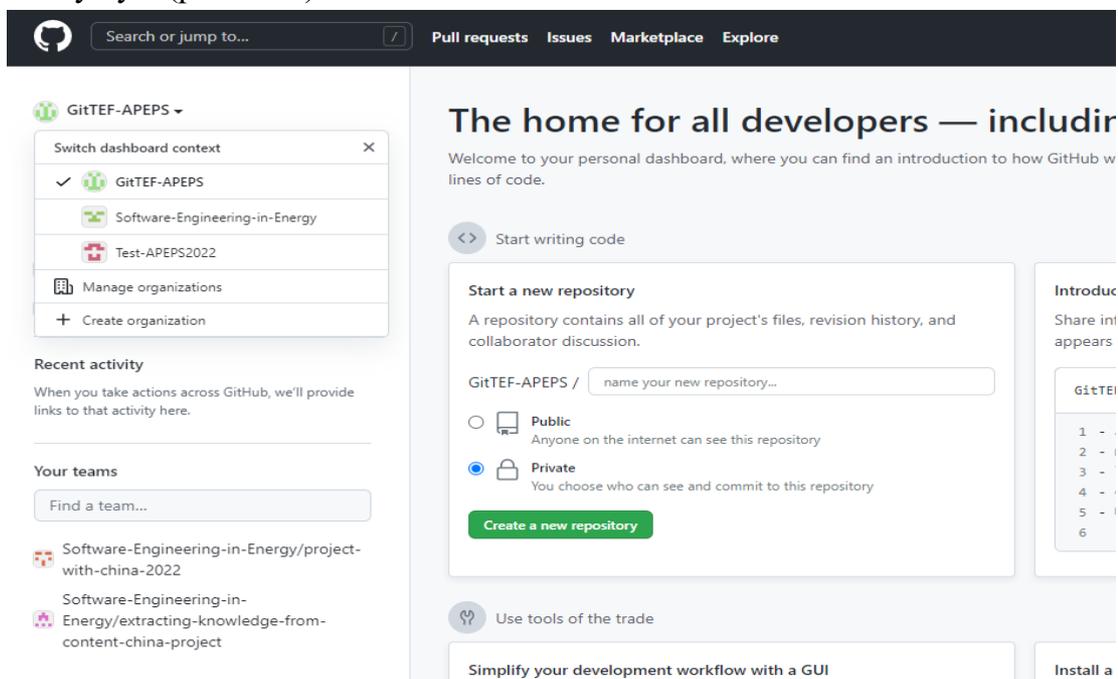


Рис. 17.7 Управління організаціями

Як і особисті облікові записи, організації безкоштовні, якщо все, що ви будете в них зберігати буде відкритим кодом.

Як власник організації, коли ви робите **fork** сховища, у вас буде вибір: робити **fork** у вашому власному просторі імен, чи у просторі імен організації. Коли ви створюєте нові сховища, ви можете створити їх або під особистим обліковим записом, або під будь-якою організацією, що її власником є ви. Також ви автоматично будете “слідкувати” (**watch**) за всіма сховищами, що ви створили для цих організацій.

Ви маєте головну сторінку організації, на якій є список усіх ваших сховищ — її можуть бачити й інші люди.

Команди

Організації можуть працювати, як з окремими людьми так і з окремими людьми через команди.

Щоб керувати командами потрібно на сторінці організації обрати і перейти за “**Teams**” (рис. 17.8).

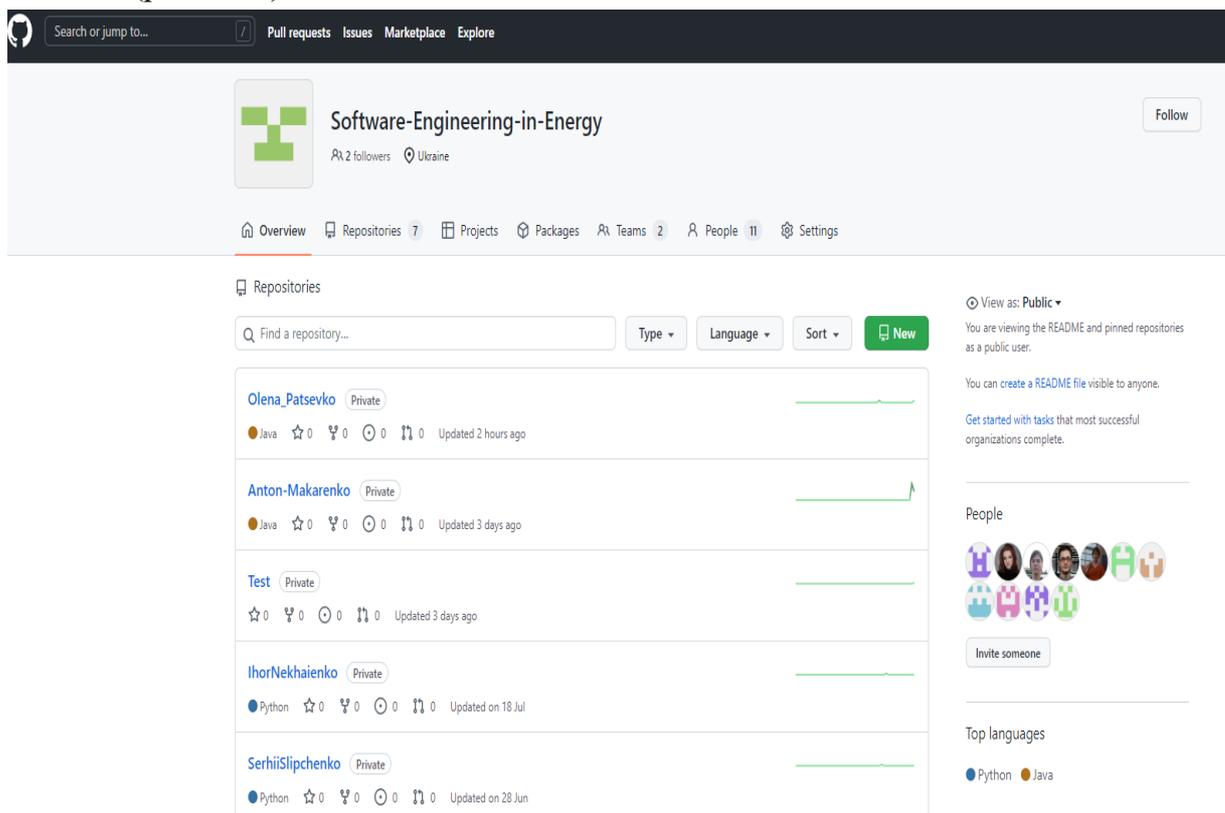


Рис. 17.8 Керування командами

Ви опинитесь на сторінці, що буде відображати команди організації та можливість створити нові – “**New team**” (рис. 17.9).

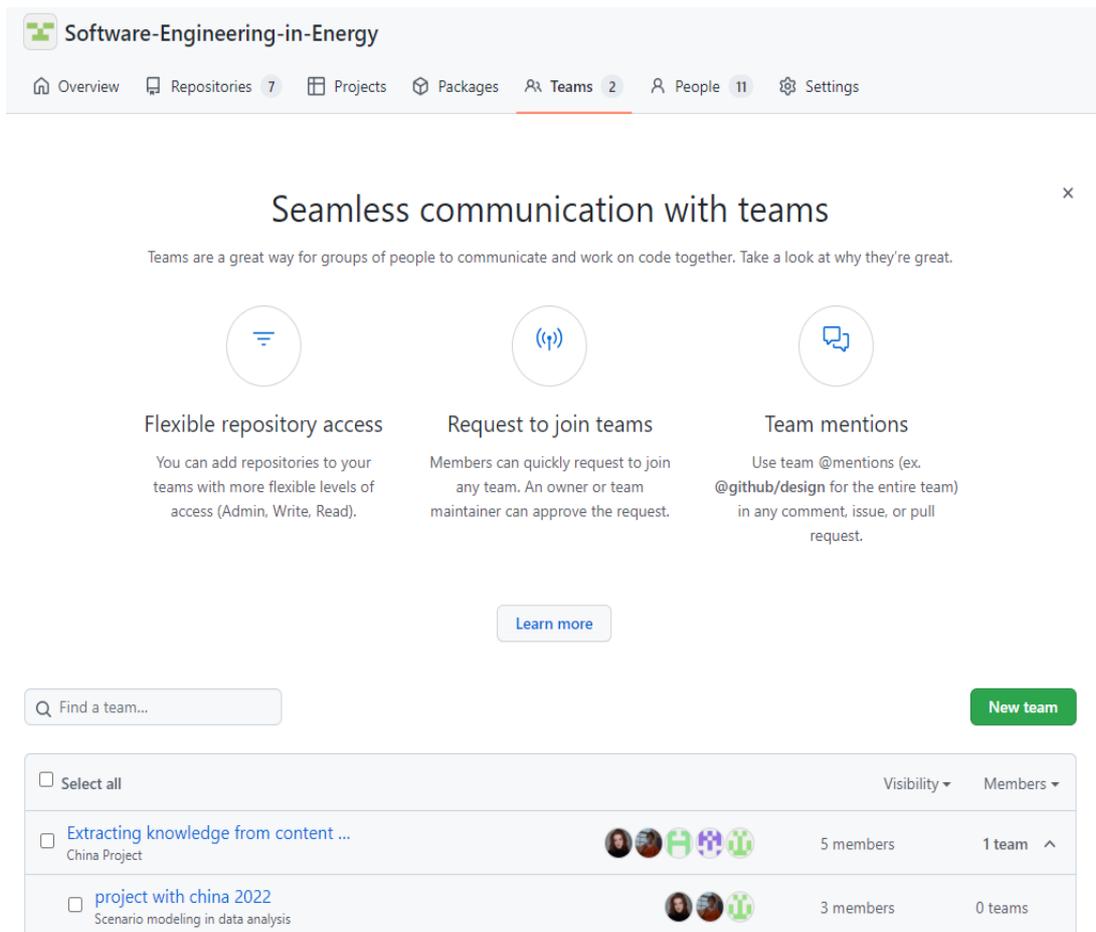


Рис. 17.9 Сторінка “Teams”

З’являється можливість приєднатися до дискусії стосовно команди (рис. 17.10).

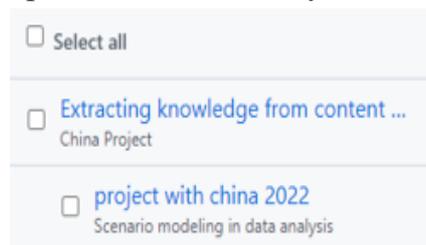


Рис. 17.10 Перехід до дискусій команд

або переглянути профілі членів команди, або створити команду – “**New team**” (рис. 17.11).

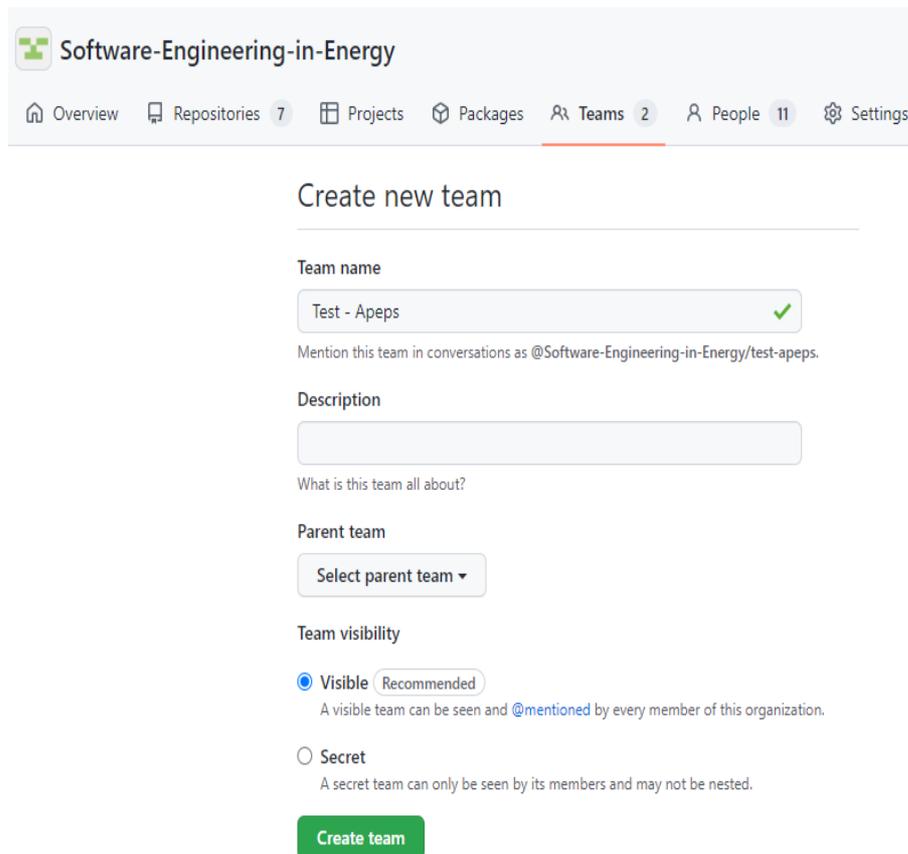


Рис. 17.11 Створення команди

Наступна сторінка після реєстрації команди відкриває можливість вести дискусію та з лівої сторони має можливість додавати “+” користувачів (рис. 17.12).

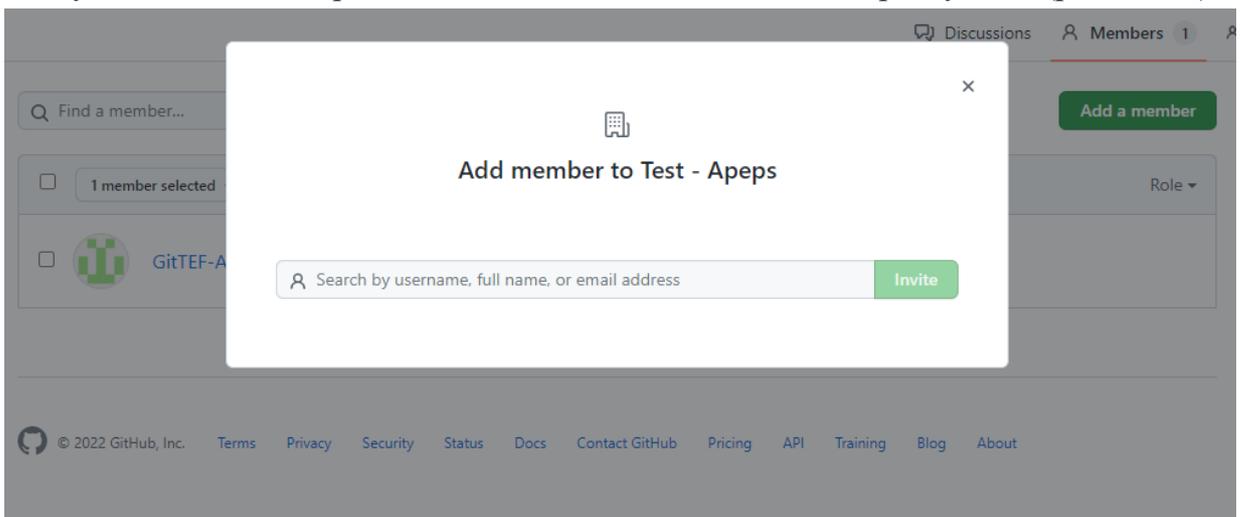


Рис. 17.12 Додавання користувачів в команди

Коли ви когось запрошуєте до команди, вони отримають листа, що повідомить їм про запрошення.

Створення репозиторію

На головній сторінці в лівій частині є можливість обрати репозиторій, або створити новий обравши  (рис. 17.3).

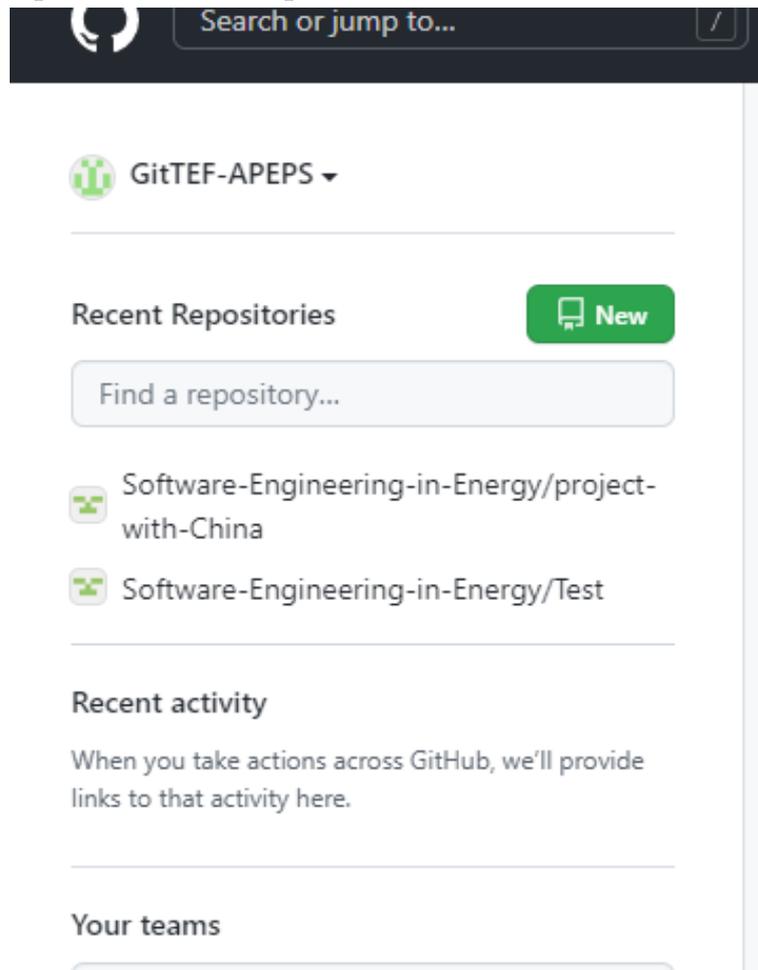


Рис. 17.13 Створення репозиторію

На новій сторінці відкривається можливість обрати показники нового репозиторію (рис. 17.4) – назву, опис (не обов’язковий), публічний чи приватний, README file, **.gitignore** (які файли будуть проігноровані при **commit**), чи є ліцензія та створити репозиторій.

Create a new repository

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

Owner * **Repository name ***

 GitTEF-APEPS ▾ /

Great repository names are short and memorable. Need inspiration? How about [turbo-bassoon?](#)

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.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

 You are creating a public repository in your personal account.

Рис. 17.14 Сторінка створення репозиторію

Ролі

Якщо увійти до організації то налаштування -  **Settings** , дають змогу перейти до сторінки де ви можете обрати  **Repository roles** (ролі в репозиторії) і ознайомитись з ролями (рис. 17.15), що призначені за замовчуванням чи створити нові ролі за вашими вимогами.

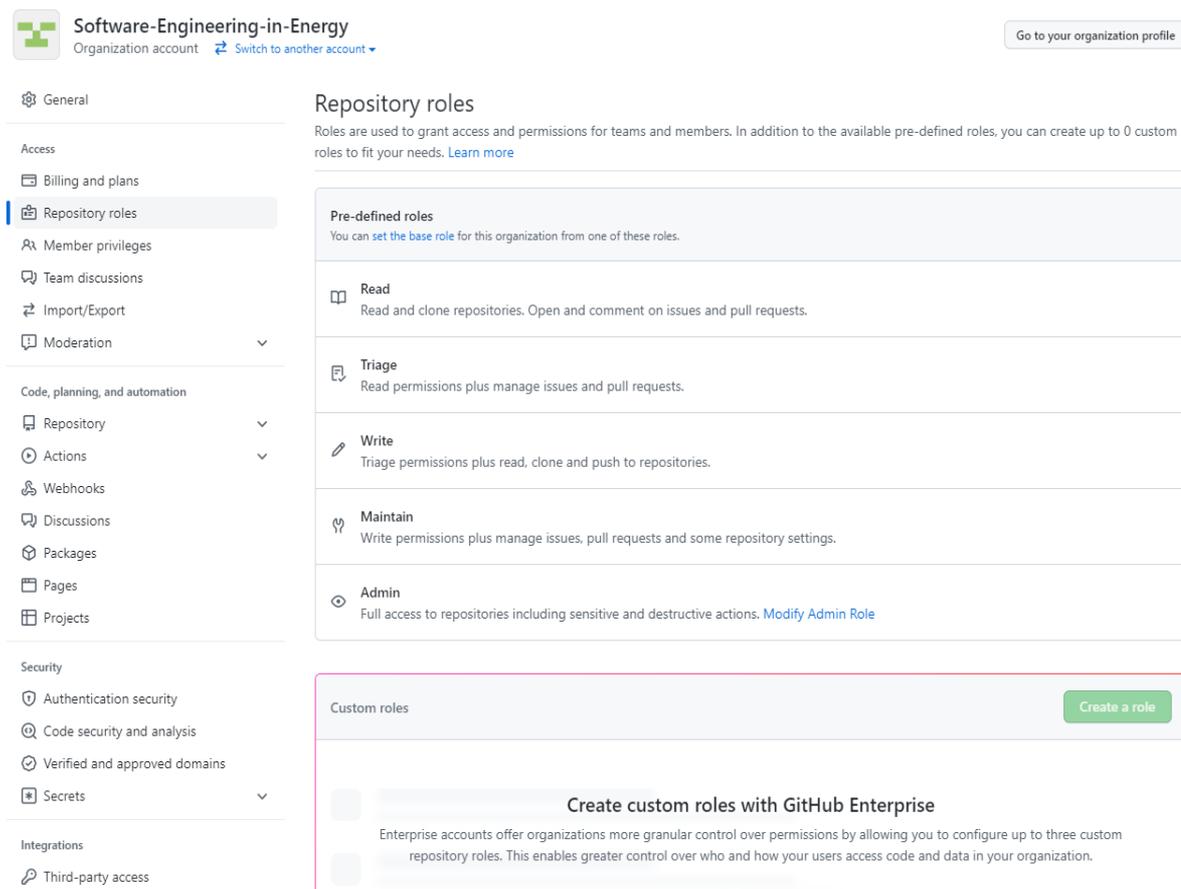


Рис. 17.15 Ролі в репозиторії

Read - Ви можете читати та клонувати сховища, відкривати та коментувати проблеми і запити на вилучення.

Triage – Ви маєте дозвіл на читання та керувати проблемами та запитами на отримання.

Write – Ви можете сортувати дозволи, а також читати, клонувати та надсилати до репозиторіїв.

Maintain – Ви маєте дозволи на запис, а також керування проблемами, запити на отримання та деякі налаштування сховища.

Admin – Ви маєте повний доступ до репозиторіїв включаючи конфіденційні та деструктивні дії. Також ви маєте можливість відредагувати роль **Admin**.

Якщо ви зайдете на сторінку певного репозиторію в організації то шляхом вибору налаштувань (рис. 17.16).



Рис. 17.16 Налаштування

ви маєте можливість переглянути учасників обравши (рис. 17.17).

Collaborators and teams

Рис. 17.17 Співробітники та команди

і опинитесь на сторінці, де можливо обрати ролі для кожного учасника (рис. 17.18 – 17.19).

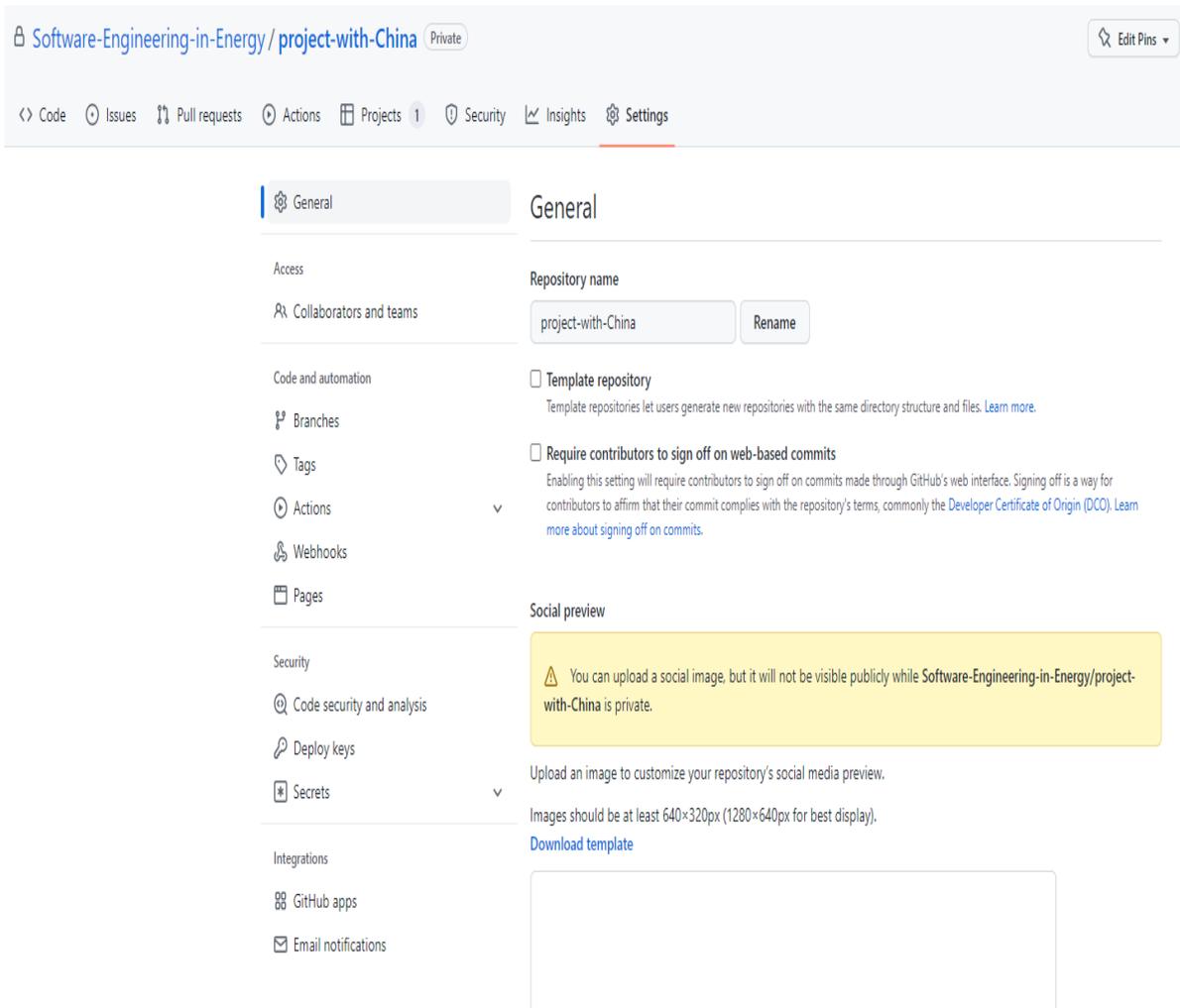


Рис. 17.18 Перехід до перегляду учасників репозиторію

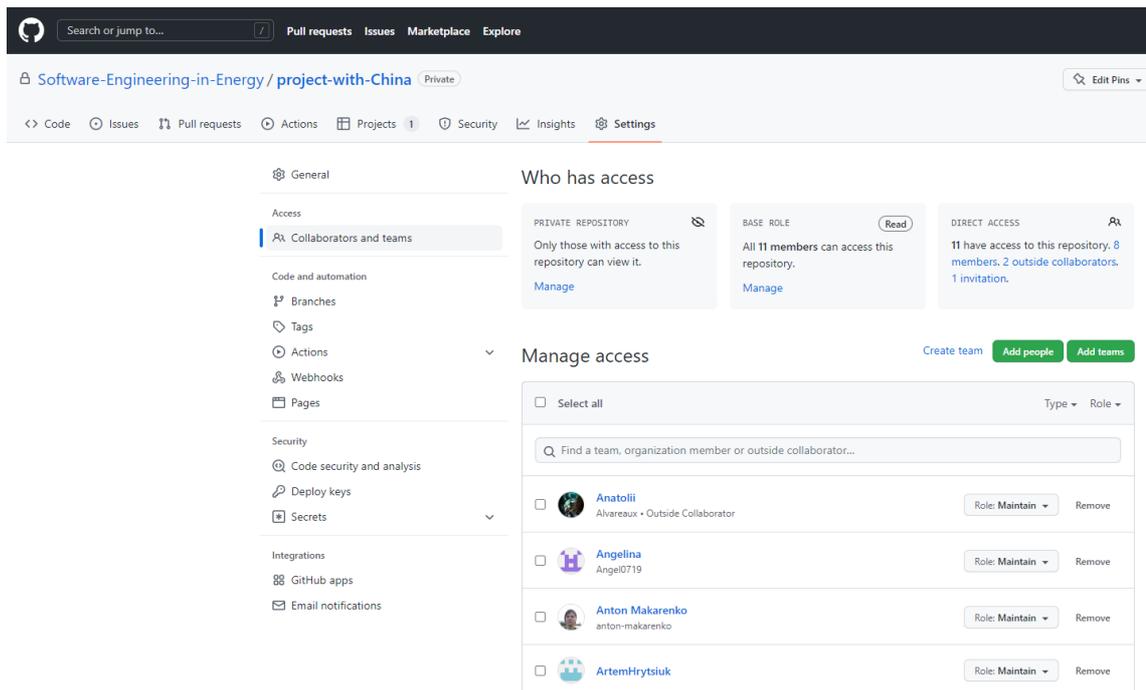


Рис. 17.19 Управління доступом учасникам репозиторію

Read - Ви можете читати та клонувати це сховище, відкривати та коментувати проблеми і запити на вилучення.

Triage (Сортування) – Ви можете читати та клонувати це сховище. Також ви можете керувати проблемами та витягувати запити.

Write – Ви можете читати, клонувати та надсилати в це сховище. Також можете керувати проблемами та витягувати запити.

Maintain – Ви можете читати, клонувати та надсилати в це сховище. Також ви можете керувати проблемами, запитами на отримання та деякими налаштуваннями сховища.

Admin – Ви можете читати, клонувати та надсилати в це сховище. Також можете керувати проблемами, запитами на отримання та налаштування сховища, зокрема додавати співавтора.

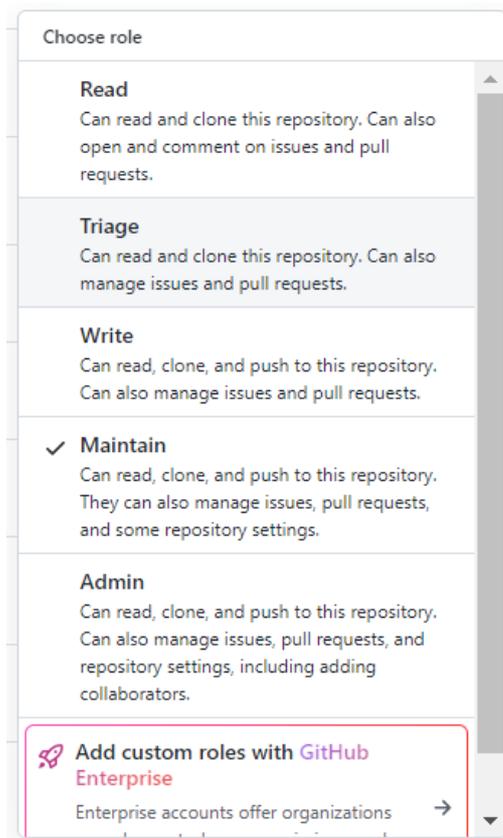


Рис. 17.20 Ролі в репозиторії

Більш детально як власники організацій можуть назначати ролі окремим особам та групам (рис. 17.20), надаючи їм різні набори дозволів в організації та як налаштувати доступ до кожного репозиторію в вашій організації, призначити детальовані ролі, надаючи людям доступ до необхідних їм функцій та задачам можна ознайомитись в наступних джерелах [18, 19].

Питання до розділу 7

1. Що таке Git та GitHub?
2. Процедура реєстрації GitHub та встановлення Git?
3. Початкові налаштування Git на локальному пристрої?
4. Налаштування за замовчуванням?
5. Організаційна основа GitHub?
6. GitHub. Опишіть організації, команди, репозиторії?
7. GitHub. Ролі в організації?
8. GitHub. Ролі в репозиторії?