



유티어(Yutier) 프로젝트 상세 배포 가이드

1. 시작하기 전에

1-1. 이 가이드의 목표

별도의 프로그램을 설치하지 않고, 인터넷 브라우저상에서 모든 코드를 수정하고 배포하는 것을 목표로 합니다.

1-2. 필수 준비

- Google(구글) 계정
- GitHub(깃허브) 계정

2. 개발 환경 준비하기

별도의 설치 과정이 없습니다. (웹 브라우저만 있으면 됩니다.)

3. 백엔드(Backend) 설정하기: Google Sheets & Apps Script

3-1. Google Sheet(구글 시트) 템플릿 준비하기

- ① '데이터베이스 템플릿' 사본을 만듭니다.
(모든 사용자의 정보(ID, 티어 등)가 저장되는 중앙 데이터베이스)

▶ <https://docs.google.com/spreadsheets/d/1pZEerJseEaWSAbWqdZeWqg93kwY7IBdYZwLhBJTrCGc/copy>



Google Sheets

문서 복사



첨부된 Apps Script 파일 및 기능도 복사됩니다.

Yuhan TRUST Administator 템플릿을(를) 복사하시겠습니까?

사본 만들기

Apps Script 파일 보기

② '매트릭스 템플릿' 사본을 만듭니다.

(각 사용자가 개인의 역량 점수를 입력하기 위해 복사해가는 양식)

▶ <https://docs.google.com/spreadsheets/d/12ymRONoIKRri2KKSiVT6SCHxJarmTsS9ZqZ-0LYt6s/copy>



문서 복사



Yuhan TRUST Matrix Template의 사본을(를) 복사하시겠습니까?

사본 만들기

3-2. '데이터베이스 템플릿' ID 확인 및 복사하기

① '3-1'의 에서 만든 '데이터베이스 템플릿'의 사본 시트를 엽니다.

② 주소창(URL)에서 /d/와 /edit 사이에 있는 ID를 복사합니다. (메모장에 붙여놓기)

The screenshot shows a Google Sheets document titled "Yuhan TRUST Administator 템플릿의 사본". The spreadsheet has a single sheet with the following data:

	A	B	C	D	E	F	G	
1	아이디	비밀번호	이름	이메일	학번	학부	전공	url
2								
3								
4								
5								
6								
7								
8								

3-3. Google Apps Script(GAS) 프로젝트 생성

① 코드 가져오기: - 유티어 원본 깃허브에 접속합니다.

▶ https://github.com/youjiyeon12/Yutier/blob/main/apps_script/backend.gs

- 화면에 보이는 코드들을 전부 복사(Ctrl+A → Ctrl+C)합니다. (약 1611줄)

main Yutier / apps_script / backend.gs

youjiyeon12 추천프로그램 세부항목 수정 842fd61 · 2 weeks ago

1611 lines (1376 loc) · 59.2 KB

Code Blame Raw ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

```

1 // Google Apps Script - Web App (GET 요청 처리)
2 // 이 스크립트는 React 프론트엔드로부터 들어오는 요청을 받아
3 // Google 스프레드시트(users, tier, major)를 읽고/쓰는 백엔드 역할을 합니다.
4 // 모든 응답은 JSON(TextOutput) 형식으로 반환합니다.
5 const SPREADSHEET_ID = '13Qf8_OnNj0FZaPkpx0_yMHz-v8ERg14lf9CXB_7bFzA'; // 실제 스프레드시트 ID로 변경
6
7 // 시트 이름 상수 정의
8 // USERS: 회원 정보, TIER: 역량 점수/합산/티어, MAJOR: 학부/전공 목록
9 const SHEET_NAMES = {
10   USERS: 'users',
11   TIER: 'tier',
12   MAJOR: 'major'
13 };
14
15 // 웹 앱 진입점: 쿼리 파라미터의 action 값으로 라우팅합니다.
16 function doGet(e) {
17   try {
18     const { action, ...data } = e.parameter || {};
19     console.log('== API 호출 시작 (GET)==');
20     console.log('요청된 액션:', action);
21     console.log('전체 파라미터:', e.parameter);
22     console.log('현재 시간:', new Date().toISOString());
23
24     // IMPORTANT: 프론트엔드에서 직렬화된 'data' 파라미터를 파싱
25     let parsedData = { ...data }; // data 객체의 속성을 복사하여 초기화
26
27     // 프론트엔드에서 직렬화된 'data' 키가 있는지 확인하고 파싱
28     if (data.data && typeof data.data === 'string') {
29       try {

```

② Google Apps Script 접속: ► <https://script.google.com> → [새 프로젝트] 클릭

☰ Apps Script

내 프로젝트

새 프로젝트

내 프로젝트

별표 표시된 프로젝트

모든 프로젝트

공유한 프로젝트

휴지통

(...) 내 실행

③ 코드 붙여넣기: '새 프로젝트' 화면의 Code.gs 내용을 다 지우고, 복사한 코드를 붙여넣습니다.

```
function myFunction() { }
```

```
// Google Apps Script - Web App (GET 요청 처리)
// 이 스크립트는 React 프론트엔드로부터 들어오는 요청을 받아
// Google 스프레드시트(users, tier, major)를 읽고/쓰는 백엔드 역할을 합니다.
// 모든 응답은 JSON(TextOutput) 형식으로 반환합니다.
const SPREADSHEET_ID = '13QT8_0nNj0FZaPpkx0_yMHz-v8ERg14lf9CXB_7bFzA'; // 실제 스프레드시트 ID로 변경

// 시트 이름 상수 정의
// USERS: 회원 정보, TIER: 역량 점수/합산/티어, MAJOR: 학부/전공 목록
const SHEET_NAMES = {
  USERS: 'users',
  TIER: 'tier',
  MAJOR: 'major'
};

// 웹 앱 전입점: 쿼리 파라미터의 action 값으로 리우어링합니다.
function doGet(e) {
  try {
    const { action, ...data } = e.parameter || {};
    console.log('*** API 호출 시작 (GET) ***');
    console.log('요청된 액션:', action);
    console.log('현재 파라미터:', e.parameter);
    console.log('현재 시간:', new Date().toISOString());
  }
}
```

④ ID 연결하기: - 코드 윗부분에서 const SPREADSHEET_ID = '...' 부분을 찾습니다.
- 따옴표 안의 내용을 '3-2'에서 메모장에 적어둔 ID로 바꿔줍니다.

```
// Google Apps Script - Web App (GET 요청 처리)
// 이 스크립트는 React 프론트엔드로부터 들어오는 요청을 받아
// Google 스프레드시트(users, tier, major)를 읽고/쓰는 백엔드 역할을 합니다.
// 모든 응답은 JSON(TextOutput) 형식으로 반환합니다.
const SPREADSHEET_ID = '13QT8_0nNj0FZaPpkx0_yMHz-v8ERg14lf9CXB_7bFzA'; // 실제 스프레드시트 ID로 변경

// 시트 이름 상수 정의
// USERS: 회원 정보, TIER: 역량 점수/합산/티어, MAJOR: 학부/전공 목록
const SHEET_NAMES = {
  USERS: 'users',
  TIER: 'tier',
  MAJOR: 'major'
};
```

⑤ 저장하기: 원하는 파일명으로 변경하고, [저장]아이콘(디스크 모양)을 누릅니다. (또는 **Ctrl+S**)



Apps Script Yutier ⚡ 저장되지 않은 변경사항

파일 A-Z + ⌛ 🔍 실행 디버그 myFunction ▾ | 실행 로그 |

Yutier.gs

```
1 // Google Apps Script - Web App (GET 요청 처리)
2 // 이 스크립트는 React 프론트엔드로부터 들어오는 요청을 받아
3 // Google 스프레드시트(users, tier, major)를 읽고/쓰는 백엔드 역할을 합니다.
4 // 모든 응답은 JSON(TextOutput) 형식으로 반환합니다.
5 const SPREADSHEET_ID = '13QT8_0nNj0FZaPpx0_yMHz-v8ERg14lf9CXB_7bFzA'; // 실제 스프레드시트 ID로 변경
6
7 // 시트 이름 상수 정의
8 // USERS: 회원 정보, TIER: 역량 점수/합산/티어, MAJOR: 학부/전공 목록
9 const SHEET_NAMES = {
10   USERS: 'users',
11 }
```

3-4. GAS 웹 앱 배포 및 주소 얻기

① [배포] > [새 배포]를 클릭합니다.



② 유형: '웹 앱'을 선택합니다.



③ 액세스 권한: '모든 사용자'로 설정하고 [배포]를 클릭합니다.



④ 액세스 승인을 진행합니다. (고급 옵션으로 이동하여 승인)

새 배포

웹 앱에서 내 데이터에 대한 액세스 권한 부여를 나에게 요청합니다.

액세스 승인

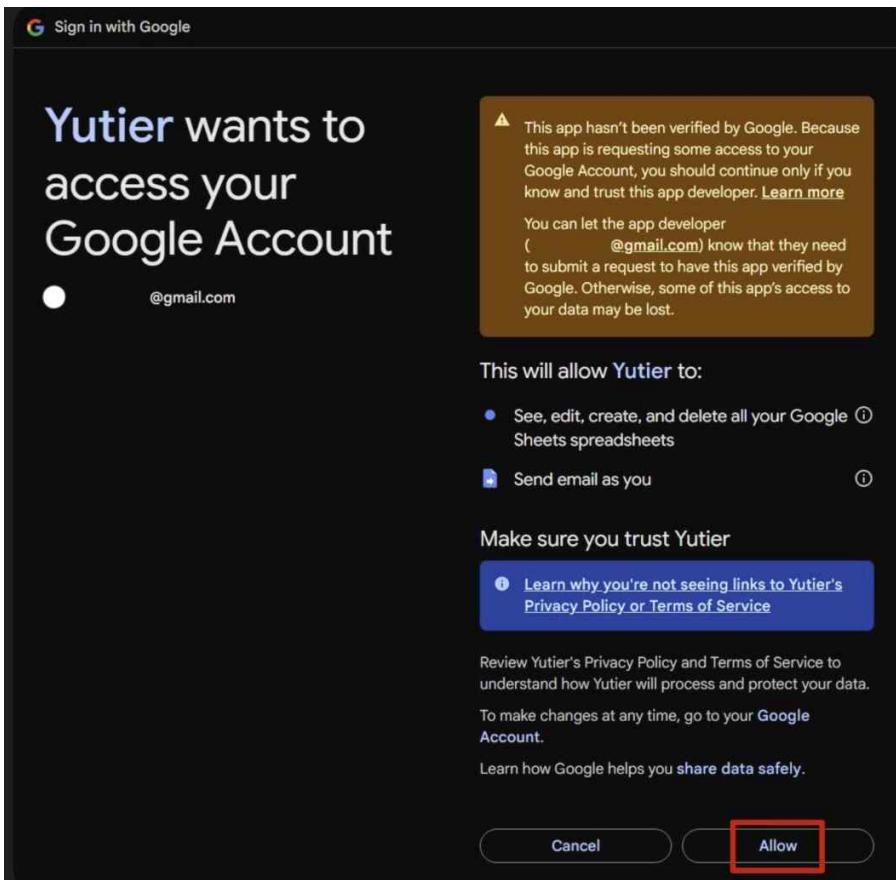


Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer
([@gmail.com](#)) verifies this app with Google, you shouldn't use it.

Advanced

BACK TO SAFETY



⑤ 배포 후 '웹 앱 URL' 복사합니다. (메모장에 붙여놓기)

배포 관리

배포가 업데이트되었습니다.

버전 (2025. 10. 23., 오후 3:35)

배포 ID

복사

웹 앱

URL

<https://script.google.com/macros/s/>

복사

라이브러리

URL

완료

4. 프론트엔드(Frontend) 설정하기

4-1. 유티어 프로젝트를 내 계정으로 가져오기(Fork)

① 유티어 원본 저장소에 방문합니다. ► <https://github.com/youjiyeon12/Yutier>

② 우측 상단의 [Fork] 버튼을 클릭하여 내 계정으로 복사합니다.

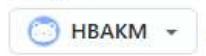
The screenshot shows the GitHub repository page for 'youjiyeon12/Yutier'. At the top right, there is a red box highlighting the 'Fork' button. Below the header, there's a navigation bar with links like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. The main content area shows a list of files and their commit history. On the right side, there are sections for 'About' (with a note: 'No description, website, or topics provided.') and 'Releases' (with a note: 'No releases published'). Below the main content, there's another navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *

/ Yutier

Yutier is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

0 / 350 characters

Copy the `main` branch only

Contribute back to [youjiyeon12/Yutier](#) by adding your own branch. [Learn more.](#)

(i) You are creating a fork in your personal account.

Create fork

4-2. 백엔드 주소 연결하기

- ① .env 파일을 업니다.

This branch is up to date with [youjiyeon12/Yutier:main](#).

[Contribute](#) | [Sync fork](#)

youjiyeon12	Update Mypage.jsx	2fbc457 · 2 days ago
📁 .github/workflows	Create deploy.yml	2 days ago
📁 apps_script	추천프로그램 세부항목 수정	last week
📁 public	Git Actions 배포 수정	2 days ago
📁 server	비밀번호 해싱 및 이전 비밀번호 재설정	3 weeks ago
📁 src	Update Mypage.jsx	2 days ago
📁 .env	추천프로그램 세부항목 수정	last week
📄 .gitignore	바로가기 배너 수정, .env 추적	last month

No description, website, or top provided.

Readme | Activity | 0 stars | 0 watching | 0 forks

Releases

No releases published | [Create a new release](#)

Packages

No packages published | [Publish your first package](#)

- ② 수정 버튼을 누릅니다.

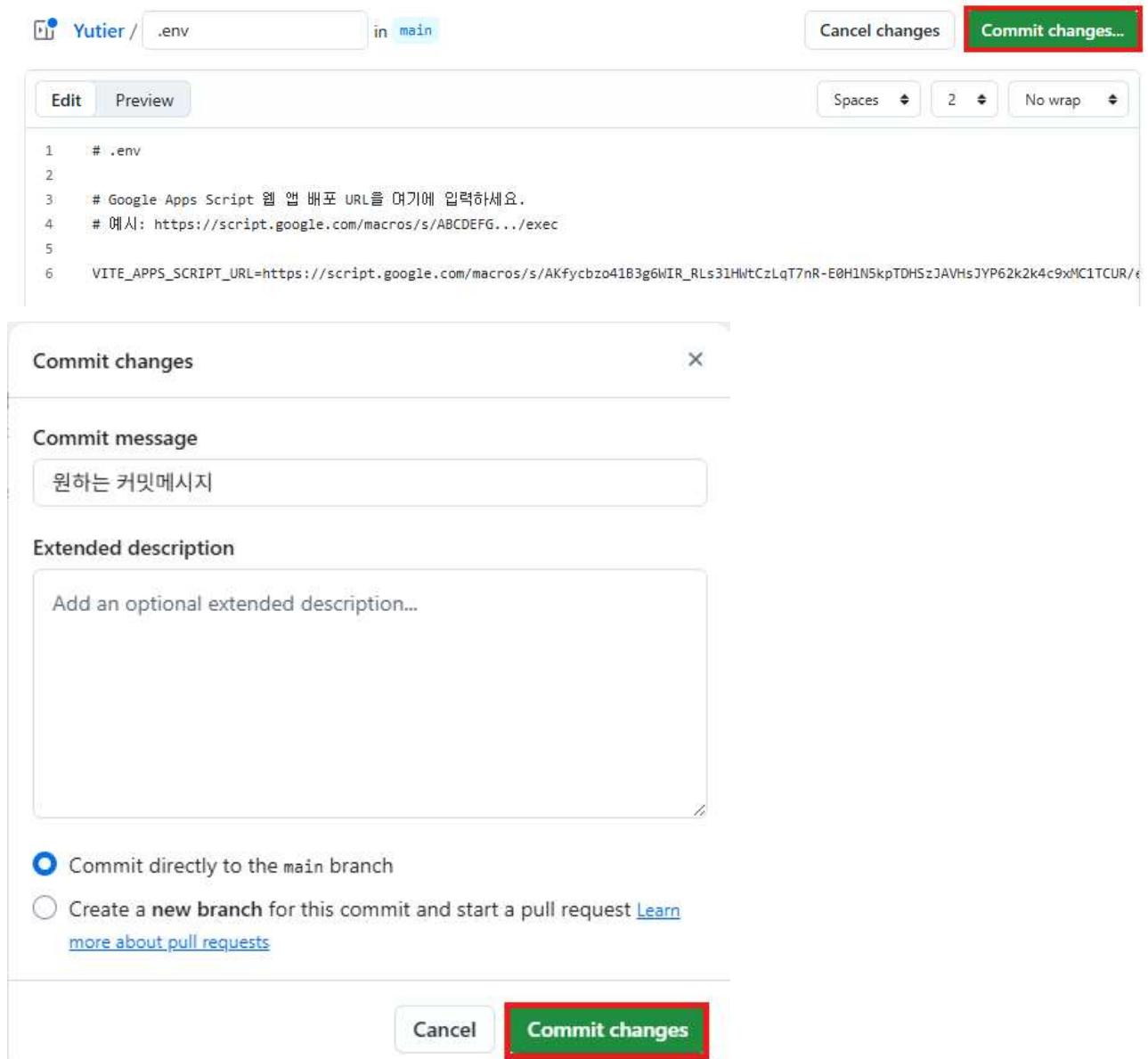
6 lines (4 loc) · 273 Bytes

Code	Blame
1 # .env	
2	
3 # Google Apps Script 웹 앱 배포 URL을 여기에 입력하세요.	
4 # 예시: https://script.google.com/macros/s/ABCDEF.../exec	
5	
6 VITE_APPS_SCRIPT_URL= https://script.google.com/macros/s/AKfycbzo41B3g6WIR_RLs31HwtCzLqT7nR-E0H1N5kpTDHSzJAVHsJYP62k2k4c9xMC1	

③ VITE_APPS_SCRIPT_URL= 부분에서 등호(=) 뒤에 공백없이 '3-4'에서 복사한 웹 앱 URL(백엔드 주소)를 붙여넣습니다.

```
1 # .env
2
3 # Google Apps Script 웹 앱 배포 URL을 여기에 입력하세요.
4 # 예시: https://script.google.com/macros/s/ABCDEFG.../exec
5
6 VITE_APPS_SCRIPT_URL=https://script.google.com/macros/s/AKfycbzo41B3g6WIR_RLs31HWEC;
```

④ Commit changes를 눌러 저장합니다.



5. GitHub Actions로 자동 배포하기

5-1. 배포 권한 설정

- 저장소에서 Settings → Pages → Source를 GitHub Actions를 선택합니다.

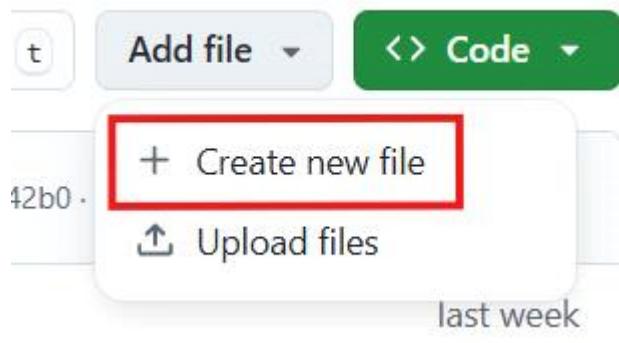
The screenshot shows the GitHub Pages settings page for a repository. On the left, there's a sidebar with various configuration tabs like General, Access, Collaborators, and Moderation options. Below that is a section for Code and automation with options for Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages. The Pages tab is highlighted with a red box. On the right, the main area is titled "GitHub Pages" and shows a summary: "Your site is live at <https://youjiyeon12.github.io/Yutier/>". Below this, under "Build and deployment", there's a "Source" dropdown menu with "GitHub Actions" selected, also highlighted with a red box. A "Configure" button is visible below the GitHub Pages Jekyll workflow card.

5-2. 배포 스크립트 만들기

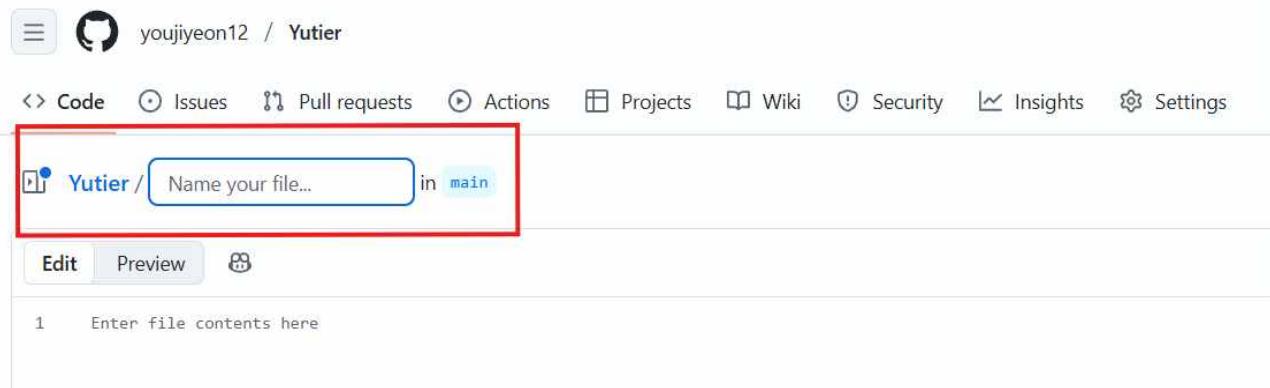
- ① 다시 [Code]탭(메인화면)으로 돌아옵니다.

The screenshot shows the main GitHub repository interface. At the top, there are several navigation tabs: Code (which is highlighted with a red box), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area is currently empty, indicating no code changes or pull requests have been made.

② [Add file] → [Create new File]버튼을 클릭합니다.



③ 파일 이름 칸에 '.github/workflows/deploy.yml' 라고 입력합니다.



④ 아래 코드를 복사해 붙여넣습니다.

```
# GitHub Actions 화면에서 보이는 워크플로 이름
name: Deploy Vite site to GitHub Pages

on:
  push:
    branches: ["main"] # main에 push할 때마다 자동배포 실행됨
  permissions:
    contents: read
    pages: write
    id-token: write
  jobs:
    build:
      runs-on: ubuntu-latest
      steps:
        - name: Checkout
          uses: actions/checkout@v4
```

```

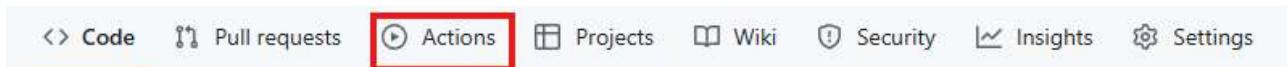
- name: Setup Node
  uses: actions/setup-node@v4
  with:
    node-version: 18
- name: Install dependencies
  run: npm ci
- name: Build
  run: npm run build
- name: Upload artifact
  uses: actions/upload-pages-artifact@v3
  with:
    path: ./dist
deploy:
  runs-on: ubuntu-latest
  needs: build
  environment:
    name: github-pages
    url: ${{ steps.deploy.outputs.page_url }}
steps:
  - name: Deploy to GitHub Pages
    id: deploy
    uses: actions/deploy-pages@v4

```

⑤ [Commit changes...] → [Commit changes] 버튼을 클릭해 저장합니다.

5-3. 배포 확인하기

① 상단 메뉴의 [Actions] 탭을 클릭합니다.



② Deploy Vite site to GitHub Pages라는 작업이 진행 중인지 확인합니다.

③ 잠시 기다리면 초록색 체크 표시로 바뀝니다. (약 1분 소요)

④ 다시 [Settings] → [Pages] 메뉴로 이동합니다.

⑤ 상단에 Your site is live at... 문구와 함께 주소가 나타납니다.

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project websites.

Your site is live at <https://youjiyeon12.github.io/Yutier/>

Last [deployed](#) by  [youjiyeon12](#) last week

-
- ⑥ 나만의 유티어 사이트가 완성되었습니다. [Visit site] 버튼을 눌러 방문해보세요.