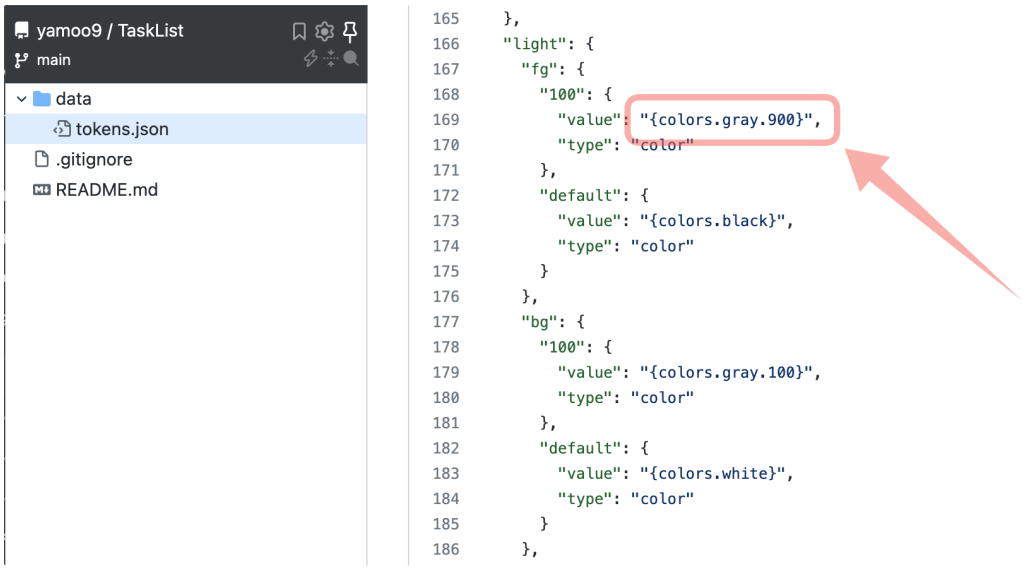
Design Token : develop process

SSAFY 특강 : UI / UX / Design Token : develop pr...

▶ 목차

토큰 변환 (Tokens Transform)

Figma Tokens 플러그인을 통해 GitHub 저장소에 푸시(push)된 토큰은 플러그인의 원시 소스(연산에 필요한 플러그인 전용 표현식 포함)라서 개발에 바로 사용할 수 없습니다. (참고)



그러므로 토큰을 개발에 사용할 수 있도록 변환(transform)해야 합니다. Figma Tokens 플러그인 개발자는 이를 위해 Token Transformer 패키지를 작성해 공개했습니다. 해당 패키지를 사용해 토큰을 변환해봅니다.

```
git clone https://github.com/yamoo9/TaskList.git cd TaskList npm i -D token-transformer
```

`token-transformer`를 사용해 토큰 입력 파일을 읽어 출력 파일을 생성하도록 명령을 실행합니다.

```
npx token-transformer 인풋파일 출력파일 병합,테마 제외테마
```

글로벌 JSON 파일 생성

Global Theme 생성 (`global.json` 파일만 생성)

```
npx token-transformer data/tokens.json theme/global.json global
```

라이트 테마 JSON 파일 생성

global + light = light Theme 생성 (`light.json` 파일만 생성)

```
npx token-transformer data/tokens.json theme/light.json global,light global
```

다크 테마 JSON 파일 생성

global + dark = dark Theme 생성 (`dark.json` 파일만 생성)

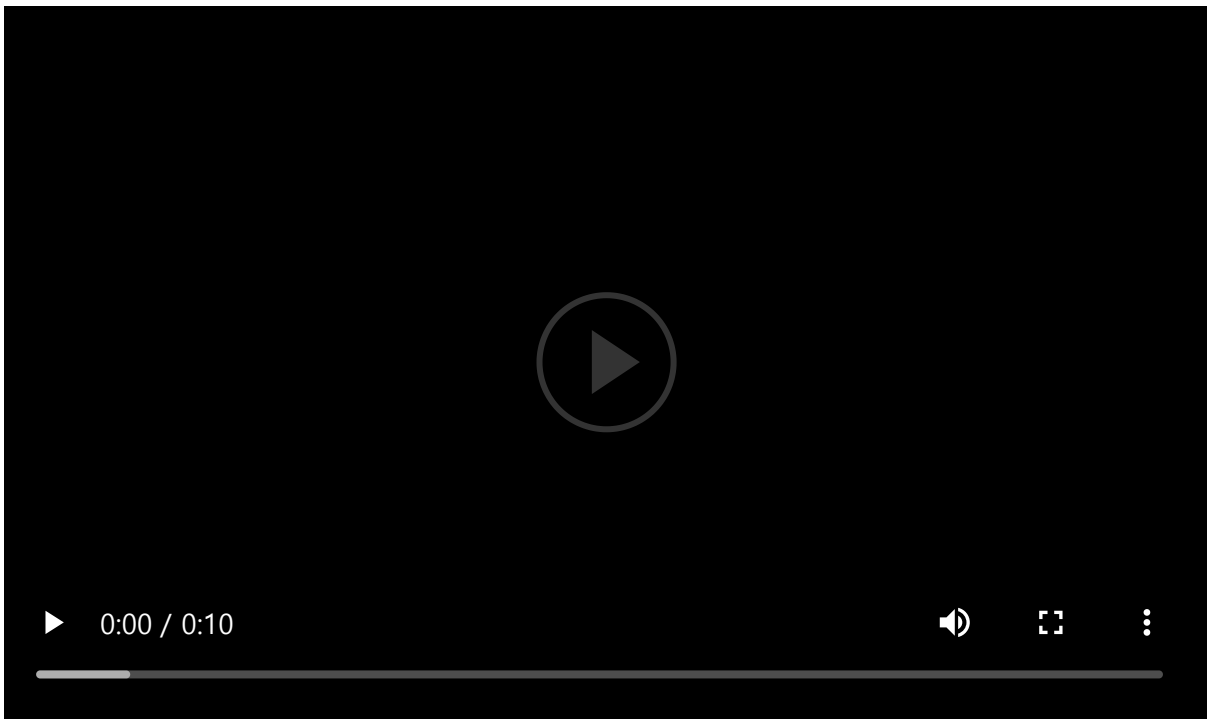
```
npx token-transformer data/tokens.json theme/dark.json global,dark global
```

명령이 실행되면 개발에서 사용 가능한 토큰 JSON 파일이 생성됩니다.

- ▶ theme/global.json
- ▶ theme/light.json
- ▶ theme/dark.json

NPM 스크립트 (NPM Scripts)

디자인이 변경되어 GitHub 저장소에 푸시(push) 될 때 마다, 매번 token-transformer 명령을 실행해 global, light, dark JSON 파일을 만드는 것은 번거롭습니다. NPM 스크립트를 등록하면 간단한 명령어로 테마 JSON 파일을 손쉽게 생성할 수 있습니다. 😉



테마 JSON 생성/제거 명령

패키지 설치 (Install Packages)

먼저 NPM 스크립트 등록에 필요한 패키지를 설치합니다. (`package.json` 파일이 없을 경우 생성합니다.)

```
# npm init -y npm i -D npm-run-all rimraf
```

명령 추가 (Add Commands)

그리고 `package.json` 파일을 열어 `scripts` 명령을 아래와 같이 추가합니다.

```
{ "scripts": { "theme:create": "run-p global light dark", "theme:delete":
"rimraf -rf theme", "global": "token-transformer data/tokens.json
theme/global.json global", "light": "token-transformer data/tokens.json
theme/light.json global,light global", "dark": "token-transformer
data/tokens.json theme/dark.json global,dark global" },
"devDependencies": { "token-transformer": "0.0.23", "npm-run-all":
"4.1.5", "rimraf": "3.0.2" } }
```

테마 생성 (Create Theme)

터미널에서 테마 JSON 파일을 생성하는 명령을 실행합니다.

```
npm run theme:create
```

테마 삭제 (Delete Theme)

생성된 테마 디렉토리를 제거할 경우, 등록된 제거 명령을 실행합니다.

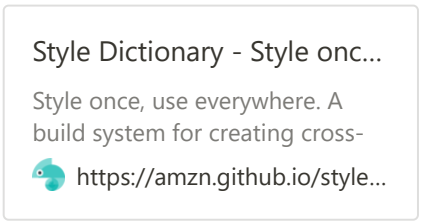
```
npm run theme:delete
```

스타일 딕셔너리 (Style Dictionary)

```
npm i -D style-dictionary
```

스타일 파일(CSS, SCSS, JavaScript) 변환

앞서 다룬 토큰 변환기(`token-transformer`)를 사용해 생성한 테마 JSON 파일을 토대로, 스타일 딕셔너리를 사용해 CSS, SCSS, JavaScript 파일을 생성할 수 있습니다.



style-dictionary 패키지를 설치합니다.

```
npm i -D style-dictionary
```

프로젝트 루트 위치에 스타일 테마 파일을 생성하는 스크립트 파일을 생성하고 코드를 작성합니다.

▶ **theme-build.js**

`package.json` 파일에 새로운 테마 생성/삭제 명령을 추가합니다.

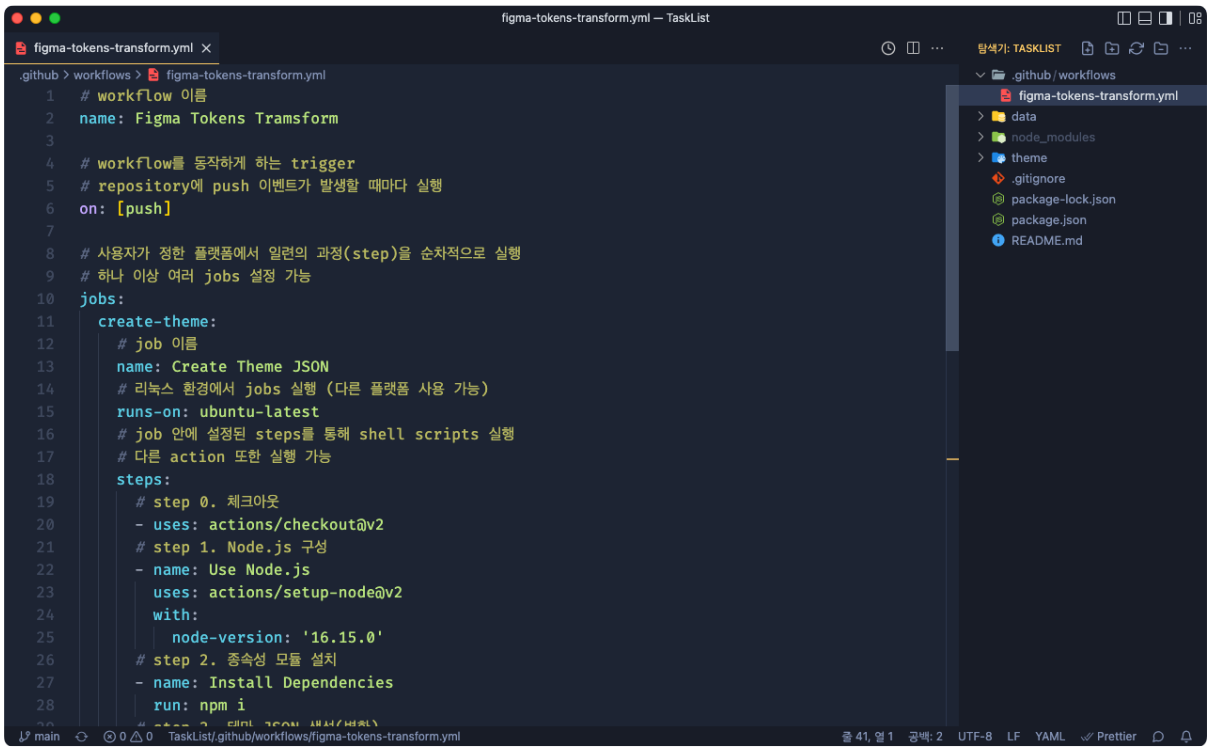
```
"scripts": { "theme:token": "run-p global light dark", "theme:create": "node theme-build.js", "theme:delete": "rimraf -rf theme/css theme/scss theme/js", "theme:delete-all": "rimraf -rf theme", "global": "token-transformer data/tokens.json theme/global.json global", "light": "token-transformer data/tokens.json theme/light.json global,light global", "dark": "token-transformer data/tokens.json theme/dark.json global,dark global" }
```

토큰 변환 자동화 (Automate Tokens Transform)

모든 것을 자동화(automation)하려면 한 단계 더 나아가야 합니다. Figma에서 토큰을 변경한 다음 GitHub 저장소에 푸시(push)하면 자동으로 토큰 JSON을 테마 JSON으로 변환(transform)하도록 구성해야 합니다.

GitHub 액션 추가 (Add GitHub Actions)

GitHub Actions을 사용해 토큰 변환을 자동 수행하도록 설정하고, 원격 저장소에 푸시(push)합니다. (참고)



▶ .github/workflows/figma-tokens-transform.yml

TaskList/figma-tokens-transform.yml at main · yamoo...

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or

https://github.com/yamoo9/TaskList/blob/main/.github/w...

yamoo9/**TaskList**

TaskList 디자인 → 개발 워크플로우

1 Contributor 0 Issues 0 Stars 0 Forks

참고

카카오웹툰은 GitHub Actions를 어떻게 사용하고 있을...

카카오엔터테인먼트 FE 기술블로그

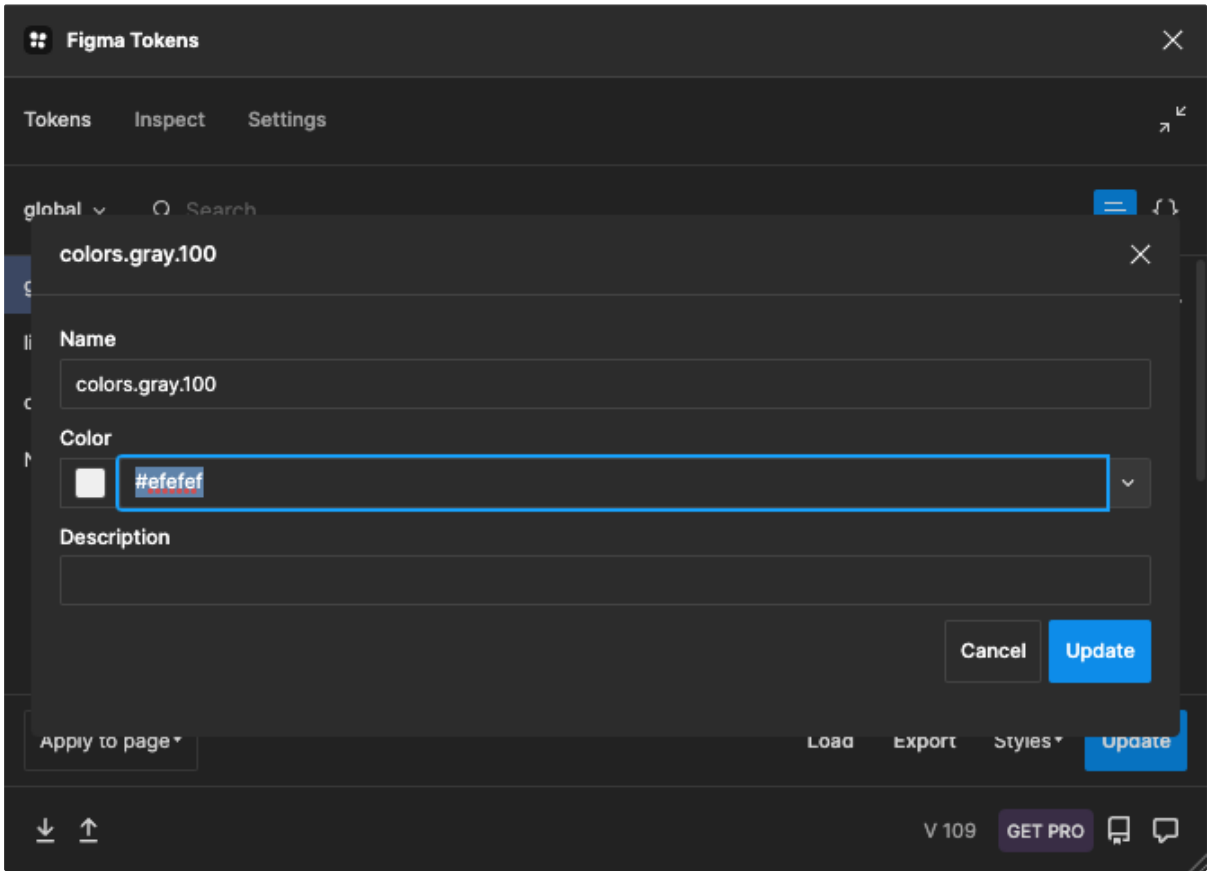
https://fe-developers.kakaoent.com/2022/220106-github-...

kakao ENTERTAINMENT

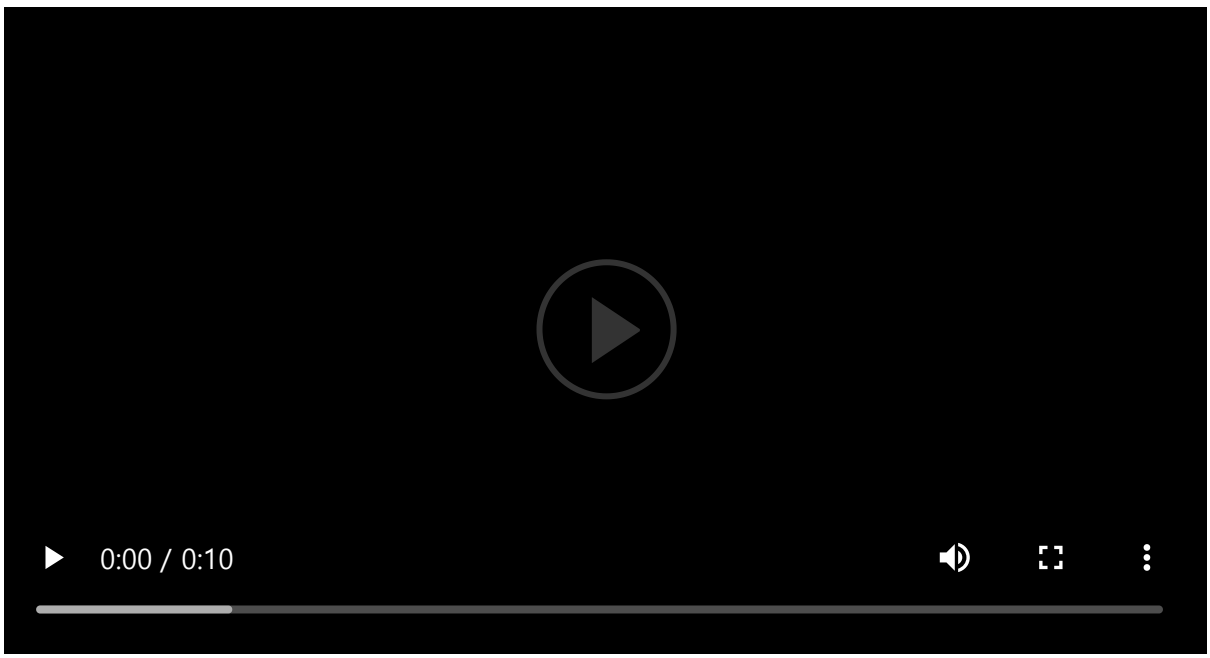
토큰 업데이트 / 푸시 (Tokens Update & Push)

등록된 GitHub Actions가 정상적으로 작동하는지 확인하기 위해 디자인 토큰을 수정하고 업데이트 합니다.

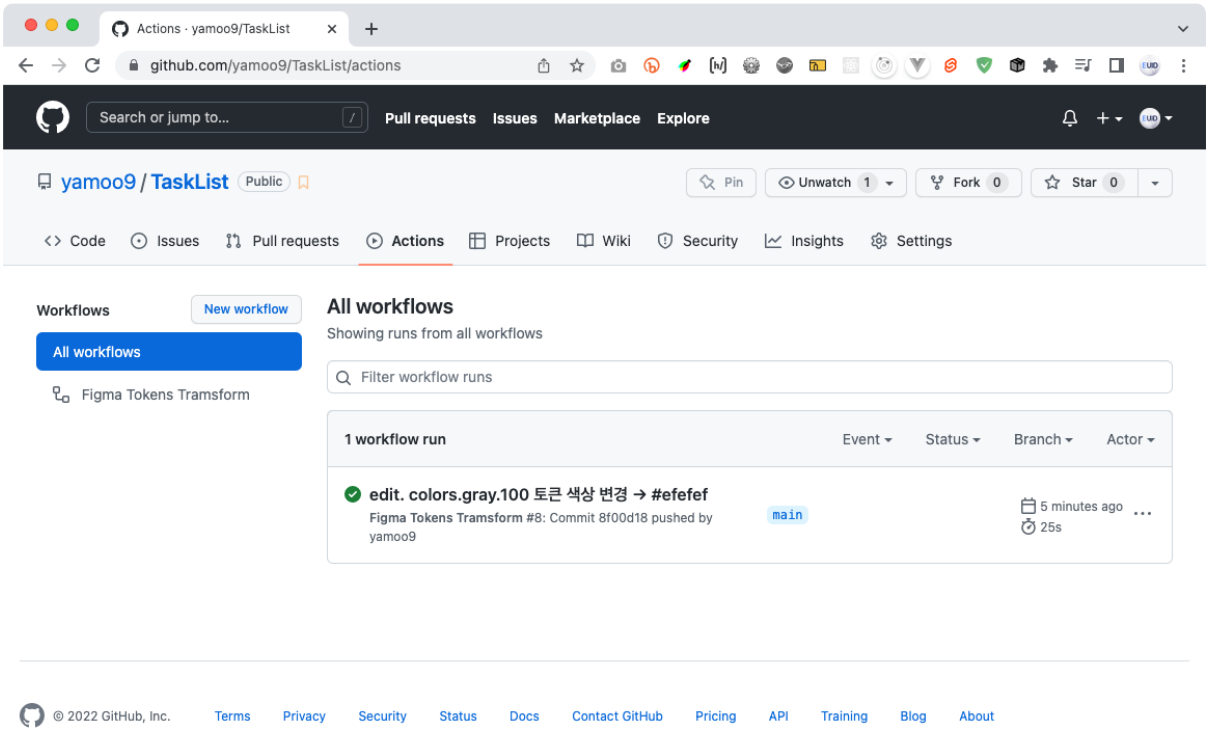
> `colors.gray.100` ~~#e6e6e6~~ → **#efefef**



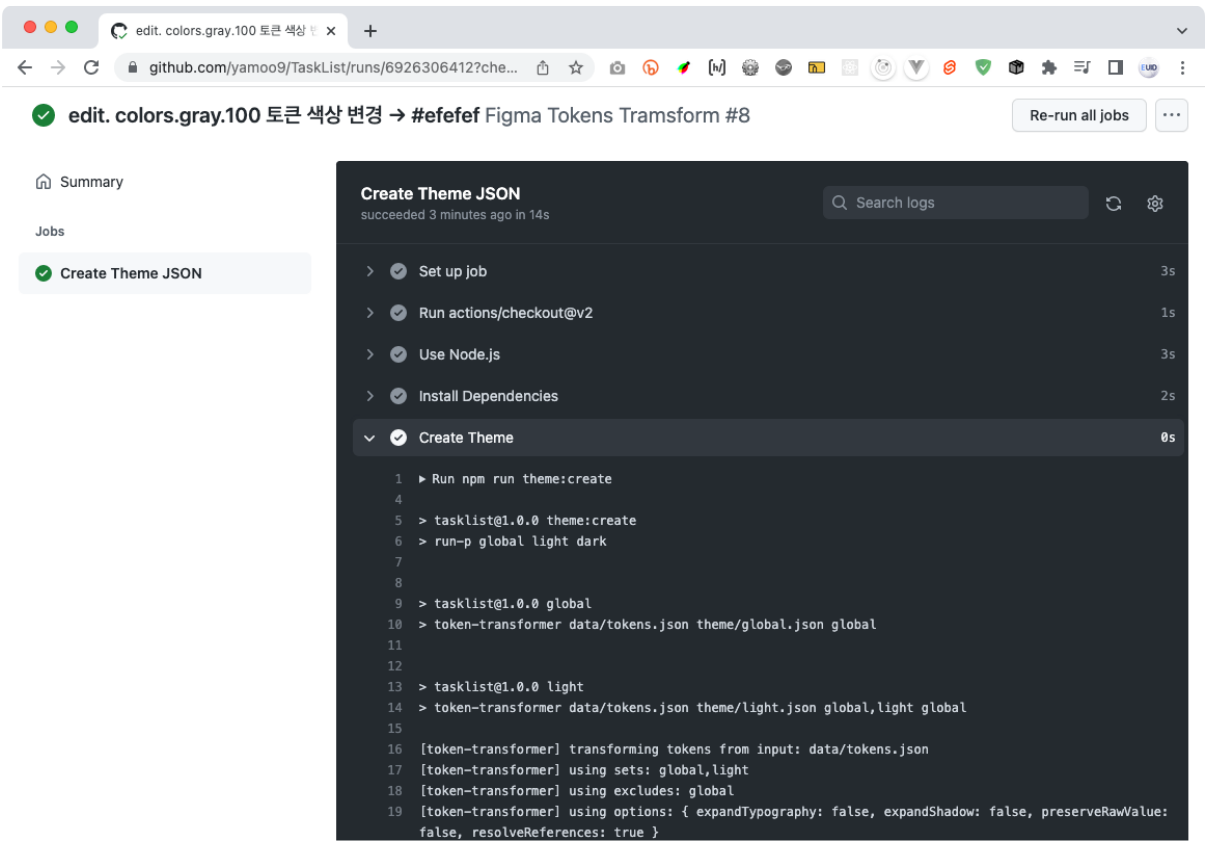
토큰을 업데이트 하면 플러그인 다이얼로그 왼쪽 하단 `Push to GitHub` 버튼에 파란색 원이 표시됩니다. `Push to GitHub` 버튼을 눌러 표시된 다이얼로그에 커밋 메시지를 작성하고 `Push` 버튼을 클릭합니다.



GitHub 저장소 → Actions 패널을 클릭하면 등록된 워크플로우가 성공적으로 실행된 것을 확인할 수 있습니다.



처리된 워크플로우를 클릭하면 상세하게 실행된 작업 과정을 살펴볼 수 있습니다.



워크플로우의 스탭 중 하나인 커밋 메시지 추가 또한 반영된 것을 커밋 리스트를 통해 확인할 수 있습니다.