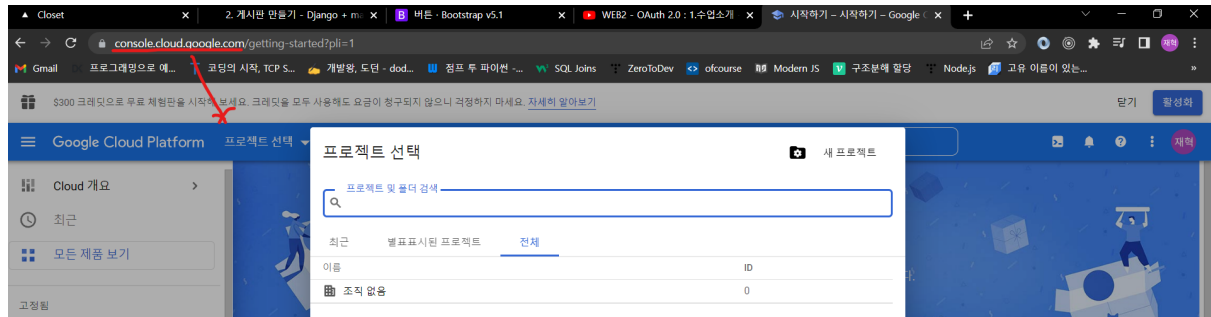
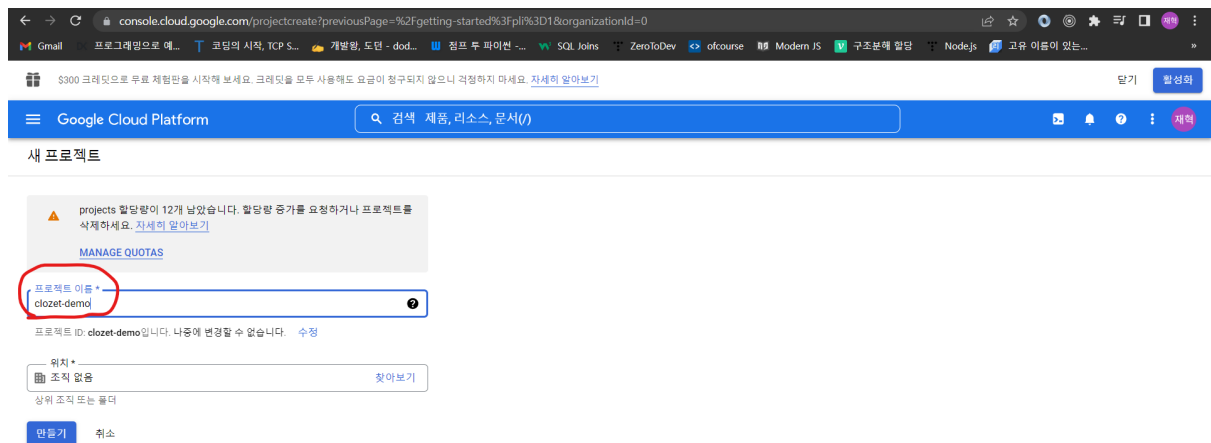


# Oauth

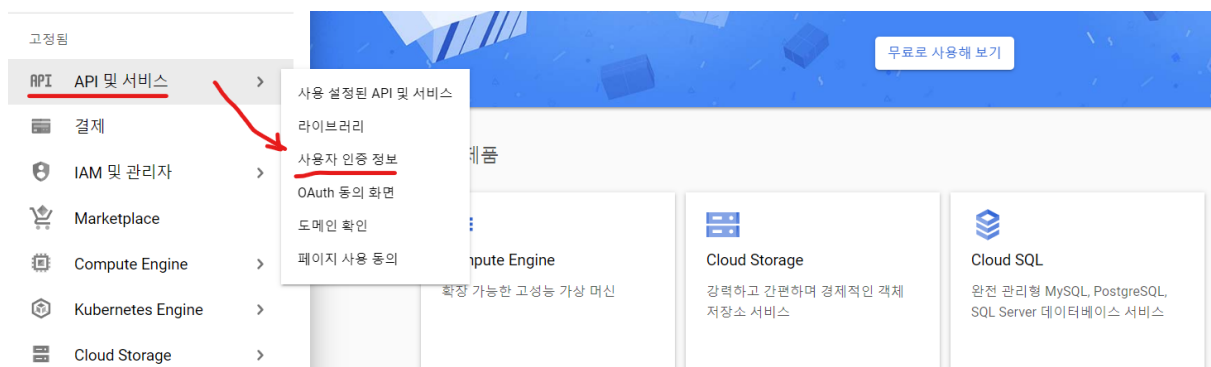
Oauth란 다양한 플랫폼 환경에서 권한 부여를 위한 산업 표준 프로토콜 입니다. 이를 통해, 인증과 권한을 획득할 수 있습니다.



1. console.cloud.google.com에서 새 프로젝트를 생성합니다.



2. 프로젝트명을 정하고, 프로젝트를 만듭니다.



3. 사용자의 인증에 대한 정보를 생성합니다. 내 프로젝트를 선택한다.



- 해당 사용자 인증 정보 만들기 메뉴에서 OAuth 클라이언트 ID를 생성합니다.  
아이디 생성을 위해서는 동의화면을 구성해야 합니다.

### OAuth 동의 화면

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

**User Type**

☐ 내부 ?

조직 내 사용자만 사용할 수 있습니다. 인증을 위해 앱을 제출할 필요는 없습니다. [사용자 유형 자세히 알아보기](#)

☐ 외부 ?

Google 계정이 있는 모든 테스트 사용자가 사용할 수 있습니다. 앱이 테스트 모드로 시작되고 테스트 사용자 목록에 추가된 사용자에게만 제공됩니다. 앱을 프로덕션에 푸시할 준비가 되면 앱을 인증해야 할 수도 있습니다. [사용자 유형 자세히 알아보기](#)

[만들기](#)

OAuth 경험에 대한 [의견 보내기](#)

- 우리 프로젝트는 배포의 목적이므로, 외부를 선택하고 만듭니다.

### 앱 등록 수정

**앱 정보**

동의 화면에 표시되어 최종 사용자가 개발자를 확인하고 문의할 수 있습니다.

앱 이름 \*  ✓


동의를 요청하는 앱의 이름

사용자 지원 이메일 \*  ✓

사용자가 동의 관련 질문을 위해 문의할 때 이용합니다.

앱 로고  ✕ [찾아보기](#)

사용자가 앱을 알아보는 데 도움이 되도록 동의 화면에 대한 이미지(1MB 이하 크기)를 업로드합니다. 허용되는 이미지 형식은 JPG, PNG, BMP입니다. 최적의 결과를 위해서는 로고가 120x120픽셀 크기의 정사각형이어야 합니다.



현재 '조직 없음' 조직의 'clozet-demo' 프로젝트를 보는 중입니다. ✕

5.

v. 앱 이름 설정

v. 사용자의 계정에 대한 질문을 받을 이메일

## V. 앱 로고 (필수는 아닙니다.)

6. 로컬 호스트에서 OAuth를 도입해보았기 때문에 애플리케이션 홈페이지에는 localhost:3000으로 작성합니다. 또한, OAuth에 대한 문의사항을 받을 개발자의 연락처를 제공합니다.

### 앱 도메인

나와 내 사용자를 보호하기 위해 Google에서는 OAuth를 사용하는 앱만 승인된 도메인을 이용할 수 있도록 허용합니다. 다음 정보가 동의 화면에서 사용자에게 표시됩니다.

애플리케이션 홈페이지

사용자에게 홈페이지 링크를 제공합니다.

애플리케이션 개인정보처리방침 링크

사용자에게 공개 개인정보처리방침 링크를 제공합니다.

애플리케이션 서비스 약관 링크

사용자에게 공개 서비스 약관 링크를 제공합니다.

#### 승인된 도메인 ?

동의 화면 또는 OAuth 클라이언트 구성에서 도메인이 사용되면 여기에서 사전 등록해야 합니다. 앱이 인증을 거쳐야 하는 경우 [Google Search Console](#)로 이동하여 도메인이 승인되었는지 확인하세요. 승인된 도메인 한도에 대해 [자세히 알아보세요](#).

+ 도메인 추가

### 앱 등록 수정

나와 내 사용자를 보호하기 위해 Google에서는 OAuth를 사용하는 앱만 승인된 도메인을 이용할 수 있도록 허용합니다. 다음 정보가 동의 화면에서 사용자에게 표시됩니다.

애플리케이션 홈페이지

http://localhost:3000/

사용자에게 홈페이지 링크를 제공합니다.

애플리케이션 개인정보처리방침 링크

사용자에게 공개 개인정보처리방침 링크를 제공합니다.

애플리케이션 서비스 약관 링크

사용자에게 공개 서비스 약관 링크를 제공합니다.

#### 승인된 도메인 ?

동의 화면 또는 OAuth 클라이언트 구성에서 도메인이 사용되면 여기에서 사전 등록해야 합니다. 앱이 인증을 거쳐야 하는 경우 [Google Search Console](#)로 이동하여 도메인이 승인되었는지 확인하세요. 승인된 도메인 한도에 대해 [자세히 알아보세요](#).

+ 도메인 추가

### 개발자 연락처 정보

이메일 주소 \*

iqeq12191219@gmail.com

이 이메일 주소는 OAuth 클라이언트 구성에서 사용될 때만 해당 이메일을 사용할 수 있습니다.

7. 이후, user가 로그인할 경우 내 웹 페이지에서 Google에게 인증 시 받을 개인 정보의 범위를 정합니다.

### 앱 등록 수정

OAuth 동의 화면

범위

테스트 사용자

요약



요약본이 나오면, 저희 앱에 대한 동의 화면 구성을 마쳤고, 언제든지 수정이 가능합니다.

이후에, 사용자 인증 정보에서 클라이언트 ID를 생성합니다.

++ 7-1. OAuth를 통한 내 웹페이지에서의 인증 및 인가의 시간은 Token의 유효 시간으로 정해집니다. 따라서, 해당 공간에 토큰 부여에 대한 설정을 제어할 수 있습니다.

## OAuth 비용 제한

### 토큰 부여 비용

토큰 부여 비용은 애플리케이션에서 새 사용자를 가져올 수 있는 속도를 제한합니다.

현재 분당 토큰 부여 비용 한도는 분당 부여 100회입니다. 분당 토큰 부여 비용은 1분마다 재설정됩니다. [일일 토큰 한도 상황](#)



[간략히 보기](#)

8. 반드시 자바스크립트 웹 어플리케이션인 것을 애플리케이션 유형에서 체크해야합니다.



12. 제 프로젝트에서는 Typescript와 Next.JS를 사용하므로, 상황에 맞는 oauth 설정이 필요합니다.

```
export const useScript = (url: any, onload: any) => {
  useEffect(() => {
    const script = document.createElement('script');

    script.src = url;
    script.onload = onload;

    document.head.appendChild(script);

    return () => {
      document.head.removeChild(script);
    };
  }, [url, onload]);
};
```

```
import { useScript } from "@/hooks";
import { RefObject, useRef, useState } from "react";

declare const google : any

export default function GoogleLogin({
  handleCredentialResponse = (response : any) => { console.log ("ENCODED JWT ID TOKEN: " + response.credential)},
}) {
  {
    useScript('https://accounts.google.com/gsi/client', () => {

      google.accounts.id.initialize({
        client_id: process.env.NEXT_PUBLIC_CLIENT_ID,
        callback: handleCredentialResponse
      })
      google.accounts.id.renderButton(
        document.getElementById('buttonDiv') as HTMLButtonElement,
        {
          theme: "outline", size: "large",
          type: 'standard', text: "signin_with"
        }
      )
    })
  }

  return (
    <div id="buttonDiv"/>
  );
};
```

Next 사용 시 브라우저를 먼저 그리지 않기 때문에, 브라우저의 기능을 사용할 수 없습니다. 따라서, useEffect를 통해서 google script를 브라우저가 그려진 후에 사용할 수 있습니다.

따라서, script에 대한 Hook을 따로 Custom하여 작성했습니다. 이후, intialize에서는 client ID를 제공해서, Token을 CredentialResponse를 통해 Return 받았습니다.

또한 아래 RenderButton은 Google에서 로그인 시 제공하는 버튼을 커스텀할 수 있습니다.