

# Google 소셜로그인

## 우리가 이미 준비해 둔 것

```
$ pip install django-allauth
```

### settings.py

```
AUTHENTICATION_BACKENDS = [  
    'django.contrib.auth.backends.ModelBackend',  
    'allauth.account.auth_backends.AuthenticationBackend',  
]  
SITE_ID = 1  
LOGIN_REDIRECT_URL = '/'
```

### urls.py

```
from django.urls import path, include  
  
urlpatterns = [  
    ....  
    path('accounts/', include('allauth.urls')),  
]
```

## provider 추가

### settings.py

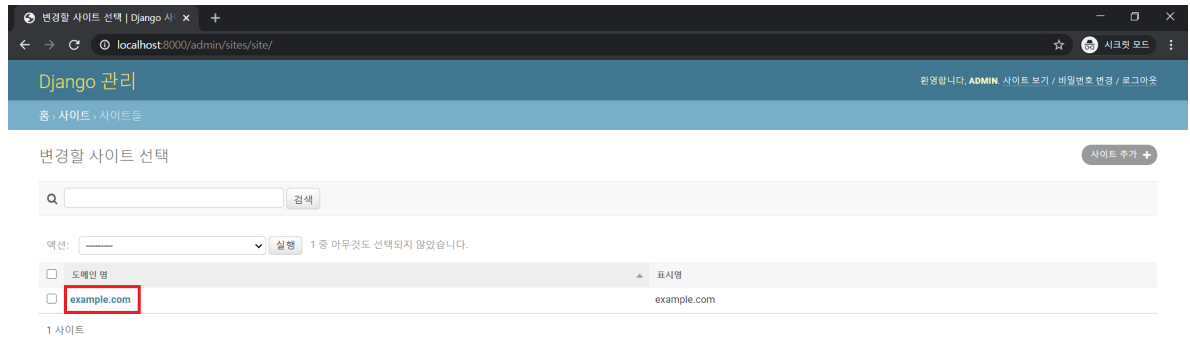
```
INSTALLED_APPS = [  
    ...  
    'django.contrib.sites',  
  
    # allauth 관련 설정  
    'allauth',  
    'allauth.account',  
    'allauth.socialaccount',  
  
    # providers  
    'allauth.socialaccount.providers.google'  
]
```

`providers` : 어떤 소셜로그인 제공업체로부터 서비스를 제공받을 것인지를 의미합니다.

## 어드민 페이지 설정

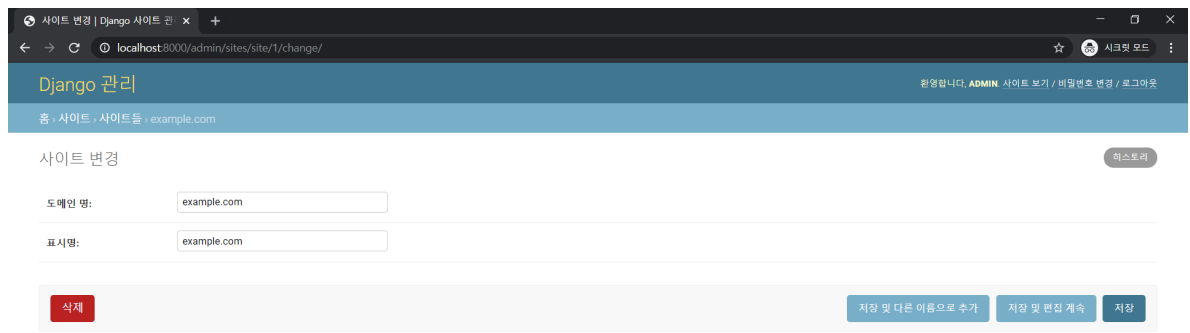


## 사이트들 클릭

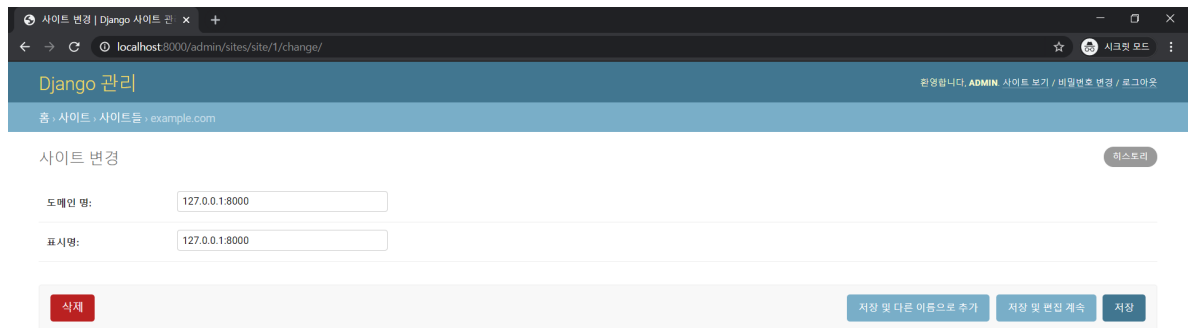


example.com 이 기본적으로 들어가 있습니다.

클릭을 하면 수정하는 페이지로 넘어갑니다.



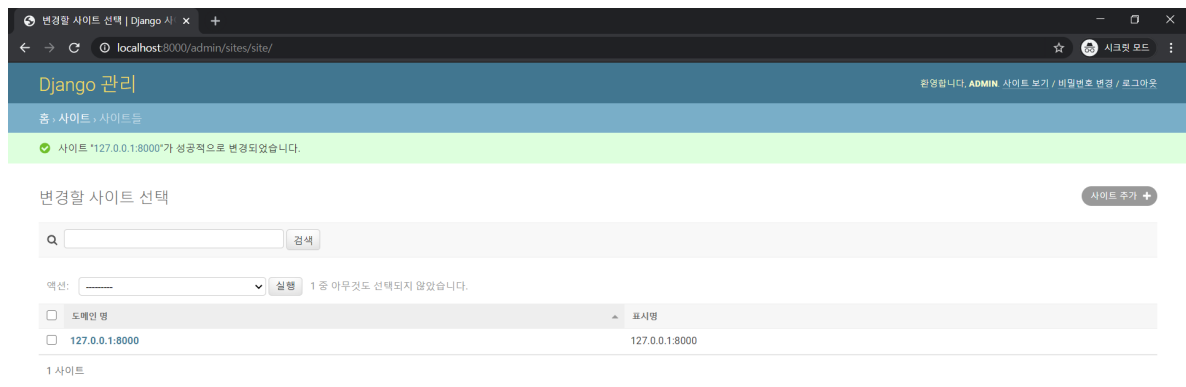
example.com으로 입력된 부분을 127.0.0.1:8000으로 수정해주세요.



여기서 127.0.0.1은 localhost 이고 8000은 포트 번호를 의미합니다.

💡 만약 나중에 프로젝트를 배포한 후 구입한 도메인이 있다면 도메인명을 입력하면 됩니다.

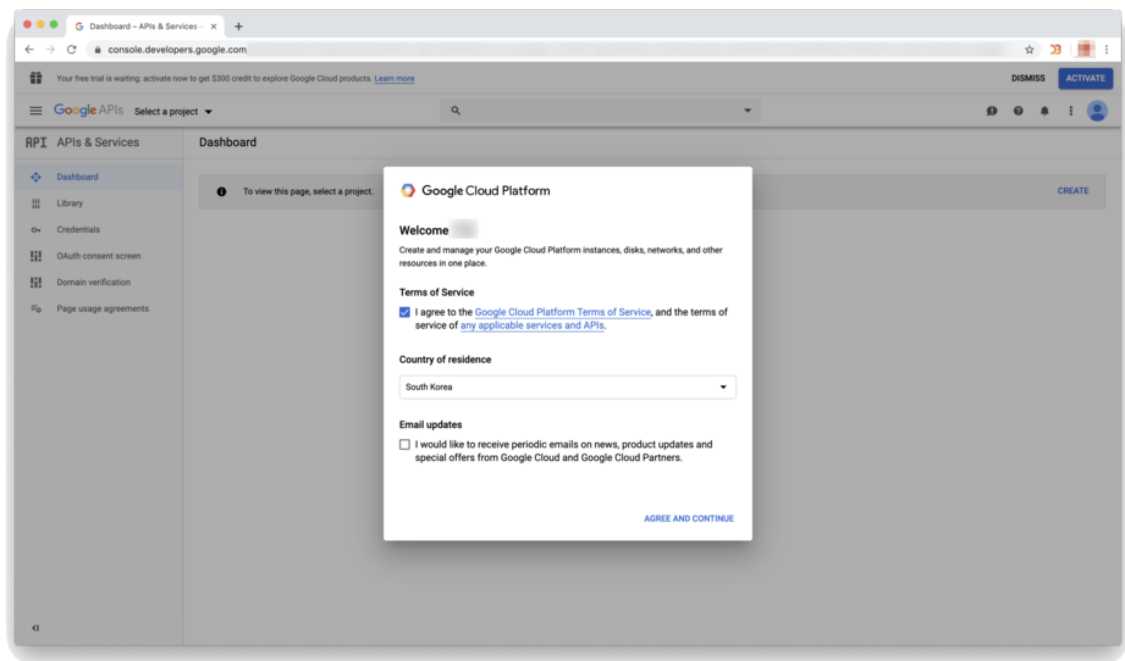
지금은 로컬에서만 작동하는 방법을 실습하겠습니다.



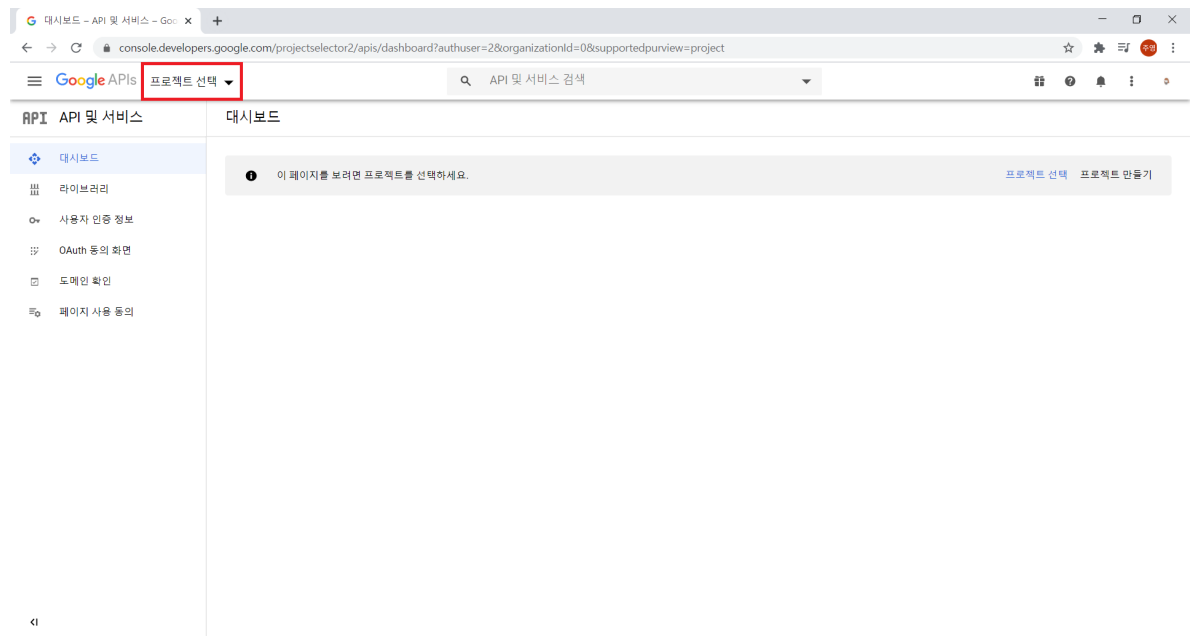
저장을 눌러주세요.

## Client id, Secret key 발급받기

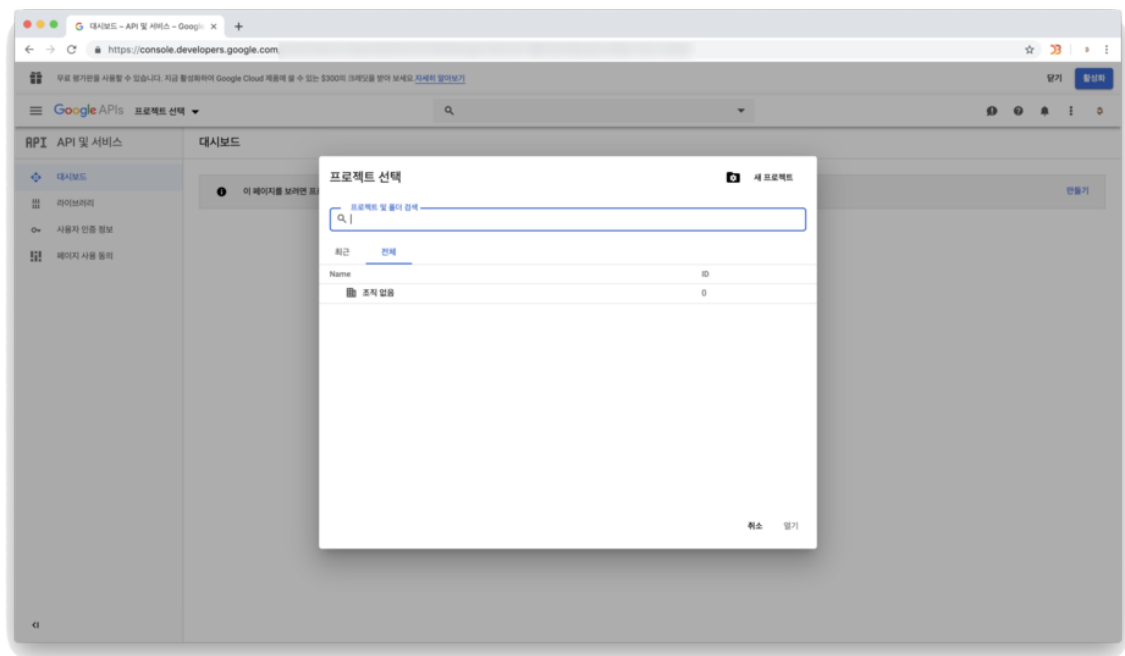
<https://console.developers.google.com/>



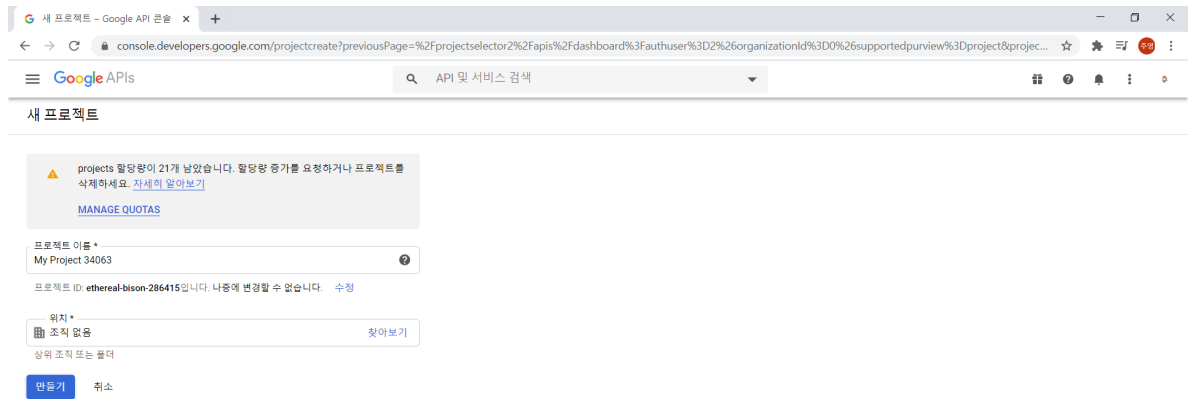
처음 접속한다면 창이 하나 뜹니다. 서비스 약관 동의, 거주지를 대한민국으로 설정한 후 확인을 눌러주세요.



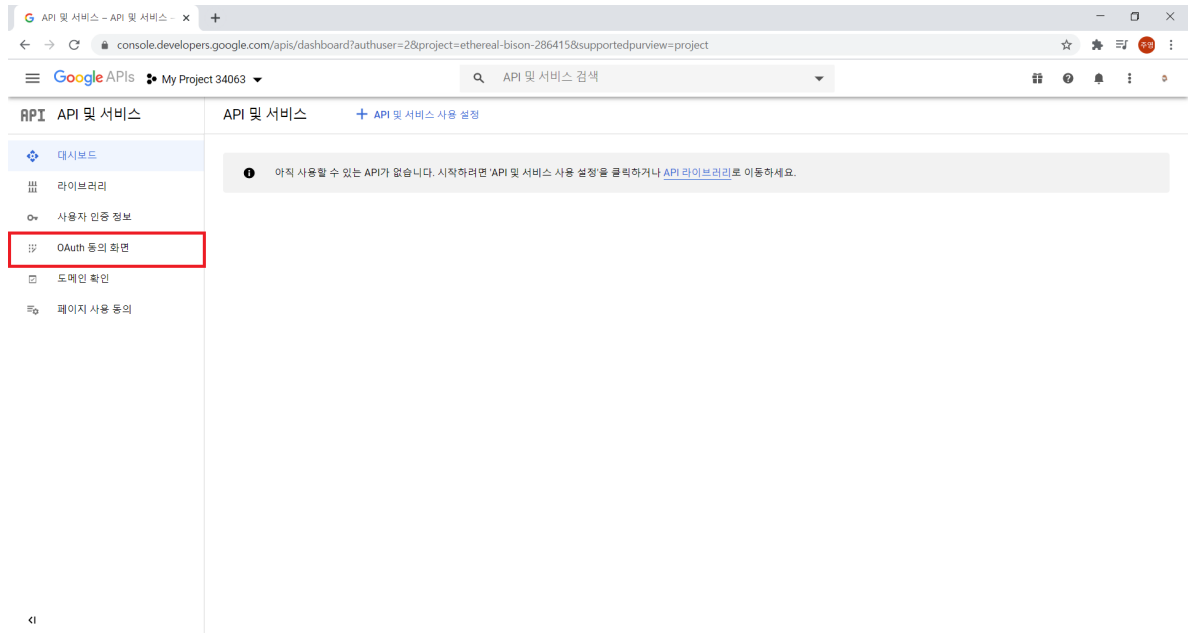
좌측 상단 **프로젝트 선택** 을 클릭해주세요.



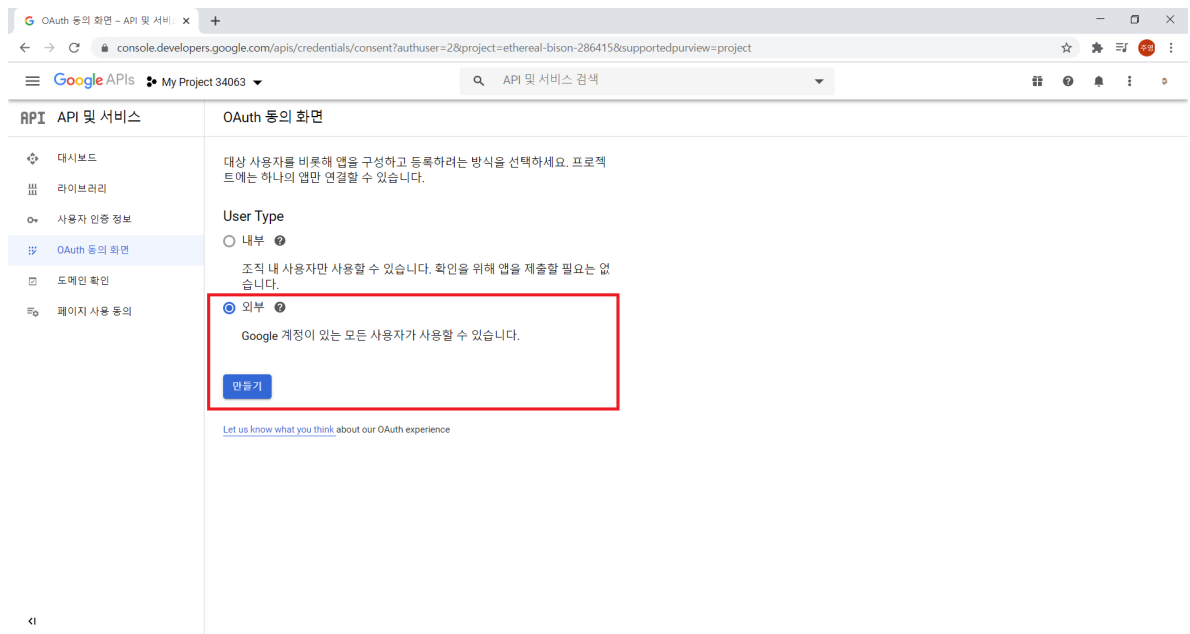
우측 상단 **새 프로젝트** 클릭



## 만들기 클릭



## OAuth 동의 화면 클릭



## 외부에 체크하고 만들기 클릭

The screenshot shows the Google API console interface for 'My Project 34063'. The left sidebar contains a list of services, with 'OAuth 2.0' selected. The main content area is titled 'OAuth 2.0' and displays the 'Authorization screen' configuration. The 'Application ID' field is highlighted with a red box. The 'Redirect URI' field is also visible. The 'Scopes' section shows the 'email' scope selected. The 'OAuth 2.0' section includes a 'Consent screen' link and a 'Test' button. The 'OAuth 2.0' section also includes a 'Consent screen' link and a 'Test' button.

여기서 입력해야 하는 건 애플리케이션 이름입니다.

The screenshot shows the Google API console interface for 'My Project 34063'. The left sidebar contains a list of services, with 'OAuth 2.0' selected. The main content area is titled 'OAuth 2.0' and displays the 'Authorization screen' configuration. The 'Application ID' field is highlighted with a red box. The 'Redirect URI' field is also visible. The 'Scopes' section shows the 'email' scope selected. The 'OAuth 2.0' section includes a 'Consent screen' link and a 'Test' button. The 'OAuth 2.0' section also includes a 'Consent screen' link and a 'Test' button.

스크롤을 내린 다음

OAuth 동의 화면 - API 및 서비스

OAuth 동의 화면

Google API의 범위  
범위를 통해 애플리케이션에서 사용자의 비공개 데이터에 액세스할 수 있습니다. 자세히 알아보기

회원만 또는 드라이브에 대한 전체 액세스 권한을 부여하는 민감한 범위를 추가하는 경우에는 Google에서 동의 화면을 확인한 후에 게시됩니다.

email

profile

openid

범위 추가

승인된 도메인  
나와 내 사용자를 보호하기 위해 Google에서는 OAuth로 인증하여 승인된 도메인을 사용하는 애플리케이션만 허용합니다. 애플리케이션의 링크가 승인된 도메인에서 호스팅되어야 합니다. 자세히 알아보기

example.com

도메인을 입력하고 Enter 키를 눌러 추가하세요.

애플리케이션 홈페이지 링크  
동의 화면에 표시됩니다. 승인된 도메인에서 호스팅해야 합니다.

https:// 또는 http://

애플리케이션 개인정보처리방침 링크  
동의 화면에 표시됩니다. 승인된 도메인에서 호스팅해야 합니다.

https:// 또는 http://

애플리케이션 서비스 약관 링크 (선택사항)  
동의 화면에 표시됩니다. 승인된 도메인에서 호스팅해야 합니다.

https:// 또는 http://

저장 | 제출하여 확인받기 | 취소

하는 방식 자세히 알아보기

OAuth 결함에 대한 의견 보내기

OAuth 부여 한도

트론 부여 비율

현재 분당 트론 부여 비율 한도는 분당 100회입니다. 분당 트론 부여 비율은 1분마다 재설정됩니다. 현재 분당 트론 부여 비율 한도는 일일 10,000회입니다. 일일 트론 부여 비율은 매일 재설정됩니다.

한도 늘리기

1시간 6시간 1월 7일 30일

이 시간 간격에 대한 데이터가 없습니다.

저장을 눌러주세요.

OAuth 동의 화면 - API 및 서비스

OAuth 동의 화면

myblog 앱 수정

확인 상태

Verification not required

Your consent screen is being shown, but your app has not been reviewed so your users may not see all of your information, and you will not be able to request certain OAuth scopes. [Learn more](#)

사용자 유형

외부

내부로 설정

OAuth 비율 제한

사용자 한도

사용자 한도는 승인되지 않은 민감하거나 제한된 범위를 요청할 때 앱에 권한을 부여할 수 있는 사용자 수를 제한합니다. 사용자 한도는 프로젝트의 전체 기간에 적용되며 재설정하거나 변경할 수 없습니다. 승인된 민감하거나 제한된 범위만 요청하는 경우에는 확인된 앱이 이 페이지에 계속 사용자 한도를 표시하지만 사용자 한도가 적용되지는 않습니다. 사용자에게 [확인되지 않은 앱 화면](#)이 표시된다면 OAuth 요청에 승인되지 않은 추가 범위가 포함되어 있기 때문입니다.

OAuth 동의 화면 - API 및 서비스

OAuth 동의 화면

myblog 앱 수정

확인 상태

Verification not required

Your consent screen is being shown, but your app has not been reviewed so your users may not see all of your information, and you will not be able to request certain OAuth scopes. [Learn more](#)

사용자 유형

외부

내부로 설정

OAuth 비율 제한

사용자 한도

사용자 한도는 승인되지 않은 민감하거나 제한된 범위를 요청할 때 앱에 권한을 부여할 수 있는 사용자 수를 제한합니다. 사용자 한도는 프로젝트의 전체 기간에 적용되며 재설정하거나 변경할 수 없습니다. 승인된 민감하거나 제한된 범위만 요청하는 경우에는 확인된 앱이 이 페이지에 계속 사용자 한도를 표시하지만 사용자 한도가 적용되지는 않습니다. 사용자에게 [확인되지 않은 앱 화면](#)이 표시된다면 OAuth 요청에 승인되지 않은 추가 범위가 포함되어 있기 때문입니다.

사용자 인증 정보 탭 클릭



사용자 인증 정보 - API 및 서비스: x

console.developers.google.com/apis/credentials?authuser=2&project=ethereal-bison-286415&supportedpurview=project

Google APIs My Project 34063 API 및 서비스 검색

API	API 및 서비스	사용자 인증 정보	+ 사용자 인증 정보 만들기	삭제
대시보드	사용 설정한 API에 액세스하려면	API 키	API 키 할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.	
라이브러리	API 키	OAuth 클라이언트 ID	OAuth 클라이언트 ID 앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.	
사용자 인증 정보	이름	서비스 계정	로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.	모든 서비스의 사용량(지난 30일)
OAuth 동의 화면	표시할 API 키가 없습니다.	OAuth 2.0 클라이언트	사용자 인증 정보 선택 도움말 사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.	
도메인 확인	이름	생성일	유형	클라이언트 ID
페이지 사용 동의	표시할 OAuth 클라이언트가 없습니다.			
	서비스 계정	이메일	이름	모든 서비스의 사용량(지난 30일)
	표시할 서비스 계정이 없습니다.			

서비스 계정 관리

## 사용자 인증 정보 만들기 > OAuth 클라이언트 ID 클릭

OAuth 클라이언트 ID 만들기 - x

console.developers.google.com/apis/credentials/oauthclient?previousPage=%2Fapis%2Fcredentials%3Fauthuser%3D2%26project%3Dethereal-bison-286415%26supportedpurview%3Dprojec...

Google APIs My Project 34063 API 및 서비스 검색

API	API 및 서비스	← OAuth 클라이언트 ID 만들기
대시보드	클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 <a href="#">OAuth 2.0 설정</a> 을 참조하세요.	<div> <div>애플리케이션 유형 *</div> <div>웹 애플리케이션</div> <div>Android</div> <div>Chrome 앱</div> <div>IOS</div> <div>TV 및 임박 제한 기기</div> <div>데스크톱 앱</div> <div>Universal Windows Platform(UWP)</div> </div>

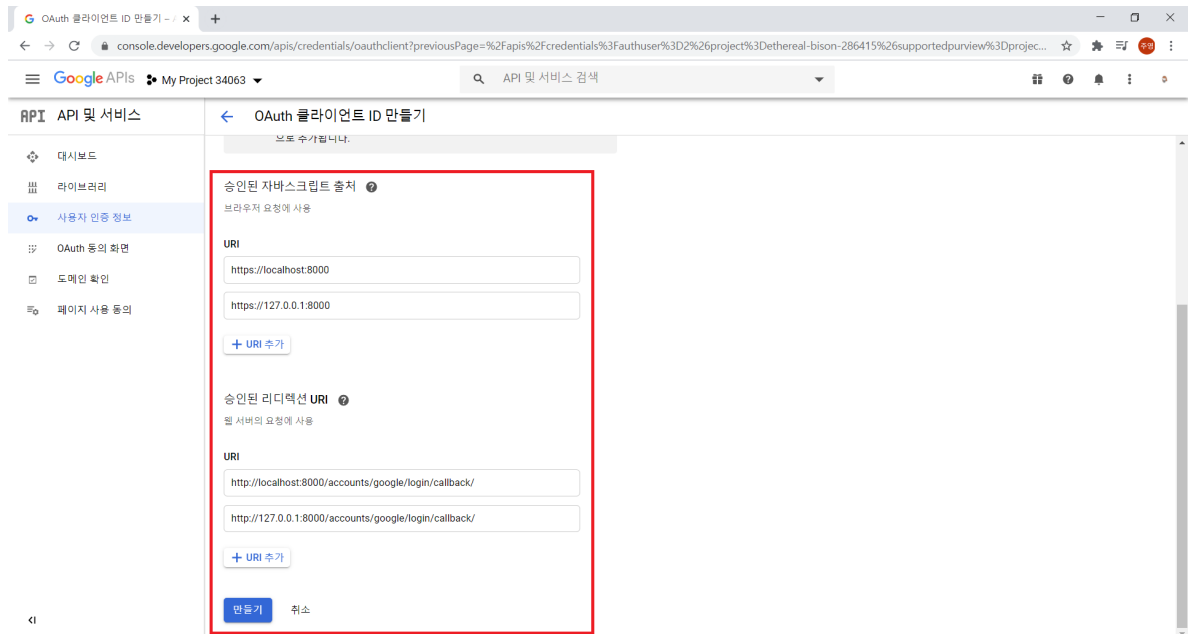
## 웹 애플리케이션 클릭

OAuth 클라이언트 ID 만들기 - x

console.developers.google.com/apis/credentials/oauthclient?previousPage=%2Fapis%2Fcredentials%3Fauthuser%3D2%26project%3Dethereal-bison-286415%26supportedpurview%3Dprojec...

Google APIs My Project 34063 API 및 서비스 검색

API	API 및 서비스	← OAuth 클라이언트 ID 만들기
대시보드	클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 <a href="#">OAuth 2.0 설정</a> 을 참조하세요.	<div> <div>애플리케이션 유형 *</div> <div>웹 애플리케이션</div> <div>OAuth 클라이언트 유형 자세히 알아보기</div> <div>이름 *</div> <div>웹 클라이언트 1</div> <div> <p>OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.</p> <p>아래에 추가한 URI의 도메인이 <a href="#">승인된 도메인</a>으로 OAuth 동의 화면에 자동으로 추가됩니다.</p> <p>승인된 자바스크립트 출처</p> <p>브라우저 요청에 사용</p> <p>+ URI 추가</p> <p>승인된 리디렉션 URI</p> <p>웹 서버의 요청에 사용</p> <p>+ URI 추가</p> </div> </div>



### 승인된 자바스크립트 출처

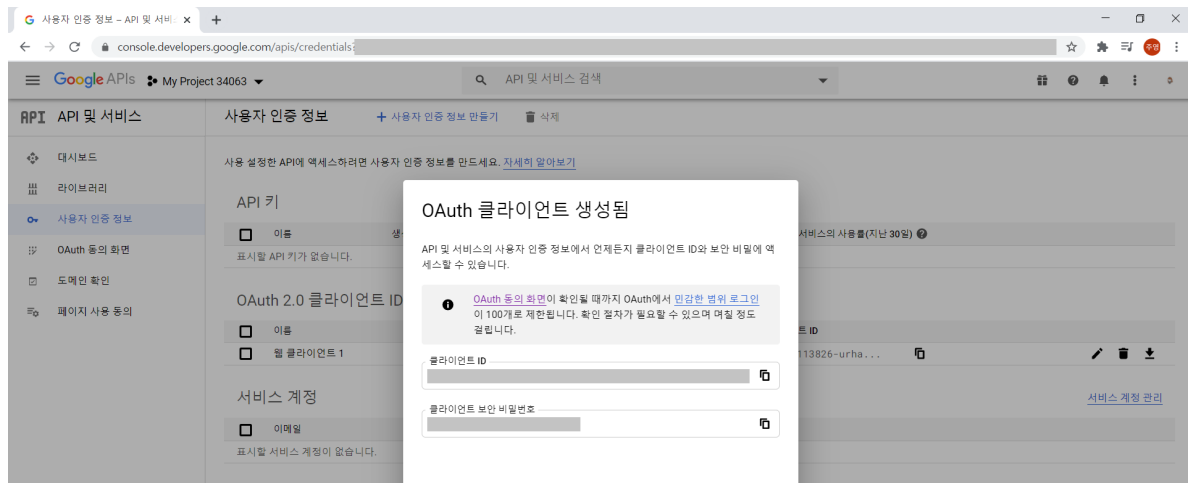
- `https://localhost:8000`
- `https://127.0.0.1:8000`

### 승인된 리디렉션 URI

- `http://localhost:8000/accounts/google/login/callback/`
- `http://127.0.0.1:8000/accounts/google/login/callback/`

💡 여기도 배포하게 되면 도메인명을 입력하세요.

📌 승인된 자바스크립트 출처는 `https` 이고, 승인된 리디렉션 URI는 `http` 로 시작한다는 점을 유의해 주세요. 다르게 작성하면 에러납니다.



이 창을 띄워놓고

(만약 실수로 꺾으면)

사용자 인증 정보 - API 및 서비스

사용자 인증 정보

API 키

OAuth 2.0 클라이언트 ID

웹 클라이언트 1

서비스 계정

이름을 클릭한 다음

웹 애플리케이션의 클라이언트 ID

이름 \*  
웹 클라이언트 1

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

아래에 추가한 URI의 도메인이 승인된 도메인으로 OAuth 등의 화면에 자동으로 추가됩니다.

승인된 자바스크립트 출처 \*  
브라우저 요청에 사용

URI  
https://127.0.0.1:8000  
+ URI 추가

승인된 리디렉션 URI \*  
웹 서버의 요청에 사용

URI  
http://localhost:8000/accounts/google/login/callback/  
+ URI 추가

클라이언트 ID

클라이언트 보안 비밀

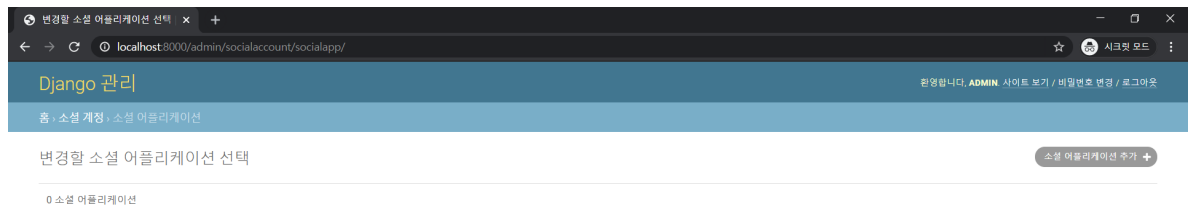
생성일  
2020년 8월 15일 AM 1시 23분 15초 GMT+9

우측에서 확인이 가능합니다.

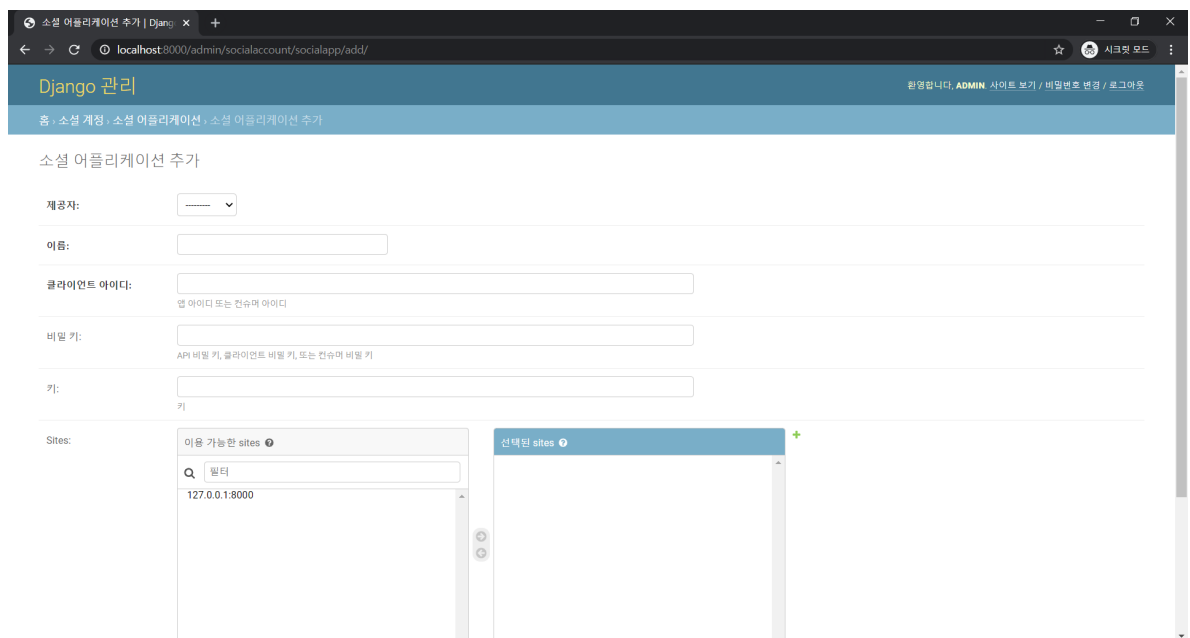
다시 어드민 메인 페이지로 돌아가서



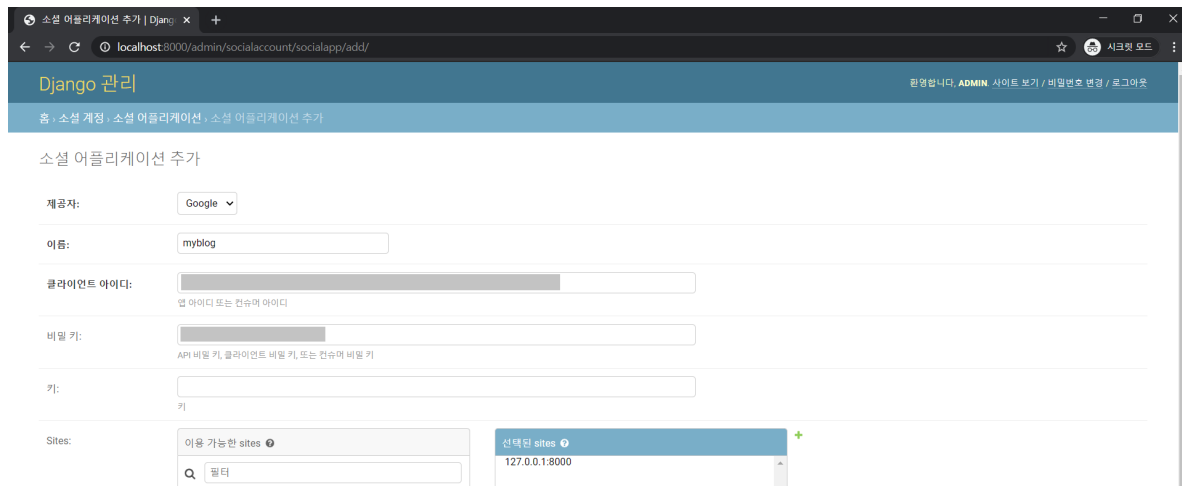
소셜 계정 에서 소셜 어플리케이션 클릭



지금은 아무것도 없음. 우측 상단에 있는 소셜 어플리케이션 추가 버튼 클릭



여기서 본인이 발급받은 제공자, 이름, 클라이언트 아이디, 비밀 키를 입력하고 **Sites** 부분에 **127.0.0.1:8000** 오른쪽으로 옮겨주세요.



Django 관리

홈 · 소셜 계정 · 소셜 어플리케이션 · 소셜 어플리케이션 추가

소셜 어플리케이션 추가

제공자: Google

이름: myblog

클라이언트 아이디:

비밀 키:

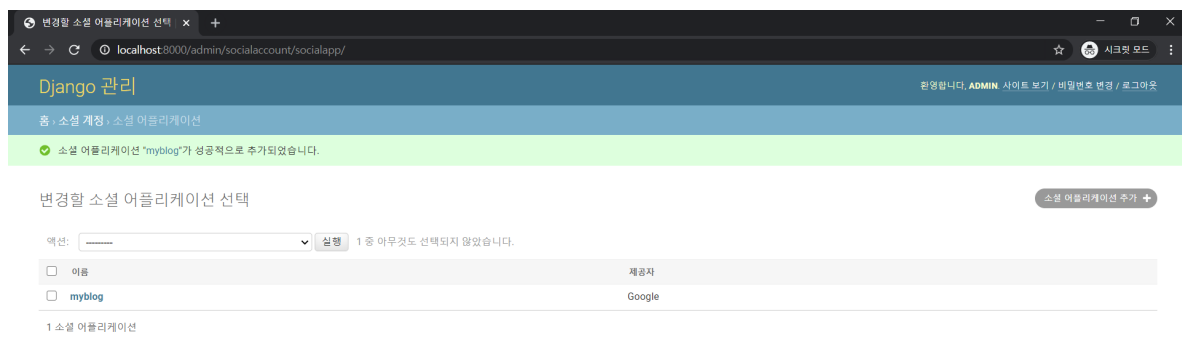
키:

Sites:

선택된 sites

127.0.0.1:8000

저장을 눌러주세요.



Django 관리

홈 · 소셜 계정 · 소셜 어플리케이션

소셜 어플리케이션 "myblog"가 성공적으로 추가되었습니다.

변경할 소셜 어플리케이션 선택

이름:

선택된 sites

127.0.0.1:8000

이름:

제공자:

myblog: Google

1 소셜 어플리케이션

잘 추가되었습니다.

## 로그인 페이지에 Google 소셜로그인 추가

`django-allauth` 의 로그인 페이지 템플릿을 가져와주세요.

<https://github.com/pennersr/django-allauth/blob/master/allauth/templates/account/login.html>

### myblog/templates/account/login.html

파일을 새로 만들어서 코드를 붙여넣겠습니다.

4번째 줄 `{% load account socialaccount %}` 아래에 `{% providers_media_js %}` 를 추가해주세요.

```
{% load account socialaccount %}
{% providers_media_js %}
```

## 로그인 페이지에 들어가보면

My Blog

← → ↻ localhost:8000/accounts/login/ ☆ 시크릿 모드

# 로그인

Please sign in with one of your existing third party accounts. Or, [sign up](#) for a 127.0.0.1:8000 account and sign in below:

- [Google](#)

또는

아이디:

비밀번호:

아이디 저장: ☐

[비밀번호를 잊으셨나요?](#)

Google 이 잘 뜨는 것을 확인할 수 있습니다.

## 클릭해보면

Google 계정으로 로그인

# 로그인

myblog(으)로 이동

이메일 또는 휴대전화

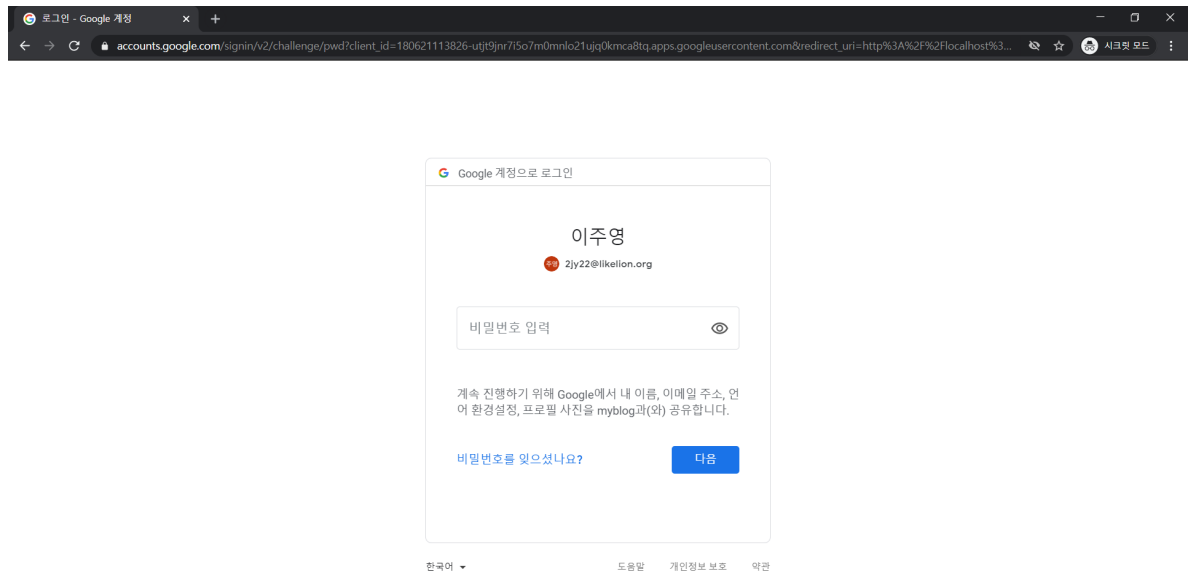
[이메일을 잊으셨나요?](#)

계속 진행하기 위해 Google에서 내 이름, 이메일 주소, 언어 환경설정, 프로필 사진을 myblog과(와) 공유합니다.

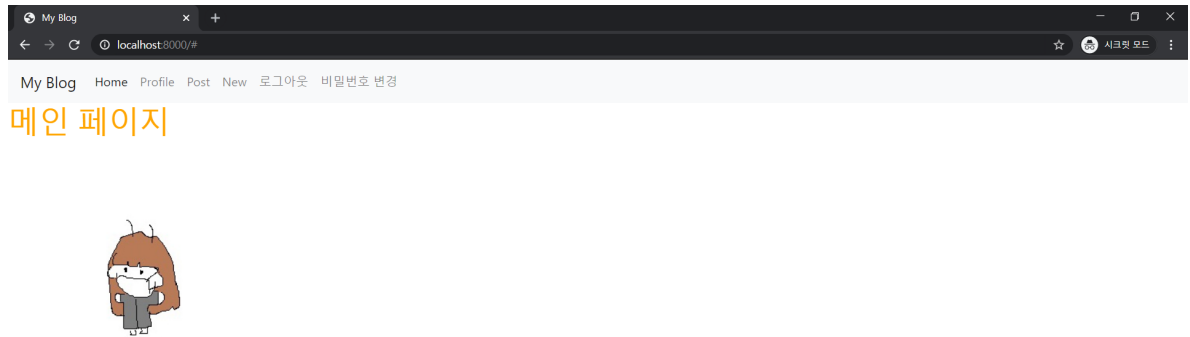
[계정 만들기](#)

한국어 ▼ 도움말 개인정보 보호 약관

로그인 창도 잘 뜹니다.

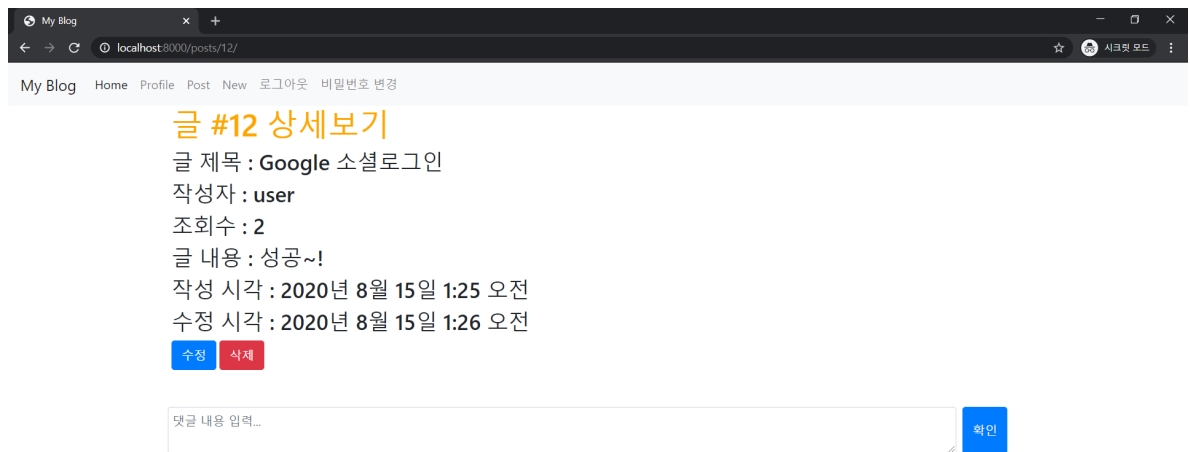


멋사 계정으로 로그인을 해보겠습니다.



로그인이 잘 됩니다!

💡 Django에서 기본으로 제공되는 `user` 모델을 이용하는 소셜로그인의 한계는 `username`의 속성으로 `user` 값이 들어가서 로그인 한 다음 글을 작성해보면 작성자에 `user`가 표시됩니다.



```
$ python manage.py shell
```

```
>>> from posts.models import *
>>> User.objects.values()
```

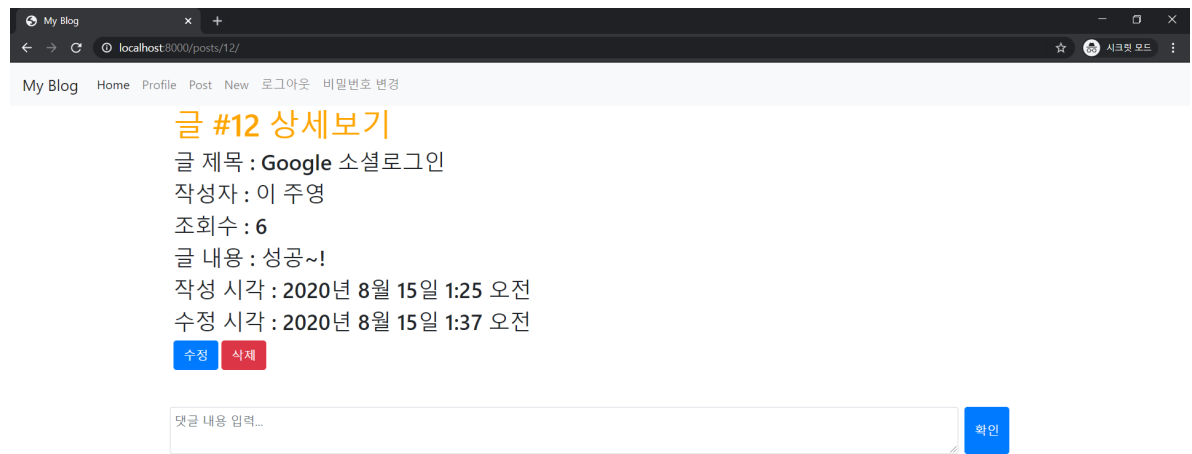
대신 기존 사용자에는 빈 값이 들어가 있던 `first_name` 과 `last_name` 에 구글 소셜로그인으로 로그인 한 사용자는 값이 들어있는 걸 확인할 수 있습니다.

## posts/templates/posts/show.html

```
{% if post.writer %}
    {% if post.writer.username == 'user' %}
        <h3>작성자 : {{ post.writer.last_name }} {{ post.writer.first_name }}
    </h3>
    {% else %}
        <h3>작성자 : {{ post.writer }}</h3>
    {% endif %}
{% else %}
    <h3>작성자 : 알 수 없음</h3>
{% endif %}
```

위와 같이 코드를 보충해주면





`user`가 아니라 Google 계정에 설정되어있는 이름으로 표시할 수 있습니다.