

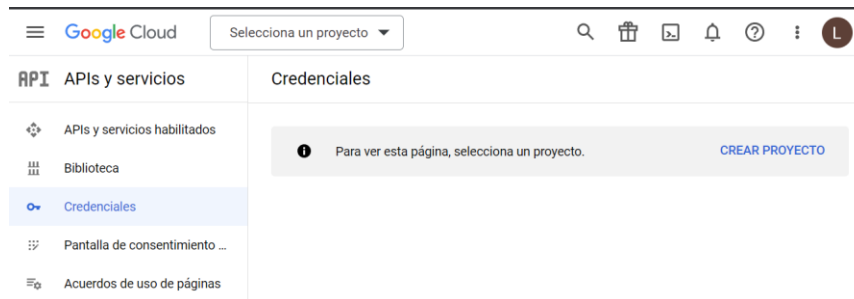
Contenido

Google Console	2
Librería	3
Configuración settings.py	4
Urls.py	5
Urls de Core.....	5
Base.html	5
Verificación	6


Google Console

- Google Console
- APIS y Servicios
- Credenciales

Crear las credenciales en Google.



Proyecto nuevo

 Tienes 12 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)


Nombre del proyecto *
TestDjango

ID de proyecto: testdjango-401219. No se podrá cambiar más tarde. [EDITAR](#)

Ubicación *
Sin organización [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

 Recuerda configurar la pantalla de consentimiento de OAuth con información sobre tu app.

[CONFIGURAR PANTALLA DE CONSENTIMIENTO](#)

Pantalla de consentimiento de OAuth

Elige cómo deseas configurar y registrar tu app, incluidos los usuarios objetivo. Puedes asociar una sola app con tu proyecto.

User Type

☐ Interno

Solo está disponible para los usuarios de tu organización. No necesitarás enviar tu app para verificarla. [Obtén más información sobre el tipo de usuario](#)

☒ Externos

Está disponible para cualquier usuario de prueba con una Cuenta de Google. Tu app se iniciará en modo de prueba y solo estará disponible para los usuarios que agregues a la lista de usuarios de prueba. Una vez que la app esté lista para enviarse a producción, puede que debas verificarla. [Obtén más información sobre el tipo de usuario](#)

[CREAR](#)

+ CREAR CREDENCIALES
BORRAR
RESTABLECER CREDENCIALES B

Clave de API

Identifica tu proyecto con una clave de API simple para verificar la cuota y el acceso

ID de cliente de OAuth

Solicita el consentimiento del usuario para que tu app pueda acceder a sus datos

Cuenta de servicio

Habilita la autenticación de servidor a servidor en el nivel de la app mediante cuentas robot

<http://127.0.0.1:8000/oauth/complete/google-oauth2/>

← Crear ID de cliente de OAuth

Un ID de cliente se usa con el fin de identificar una sola app para los servidores de OAuth de Google. Si la app se ejecuta en varias plataformas, cada una necesitará su propio ID de cliente. Consulta [Configura OAuth 2.0](#) para obtener más información. [Obtén más información](#) sobre los tipos de clientes de OAuth.

Tipo de aplicación *

Aplicación web

Nombre *

Mi Restaurante

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu [pantalla de consentimiento de OAuth](#) como [dominios autorizados](#).

Orígenes autorizados de JavaScript

Para usar con solicitudes de un navegador

+ AGREGAR URI

URI de redireccionamiento autorizados

Para usar con solicitudes de un servidor web

URI 1 *

<http://127.0.0.1:8000/oauth/complete/google-oauth2/>

+ AGREGAR URI

Se creó el cliente de OAuth

Puedes acceder al ID de cliente y el secreto desde "Credenciales" en API y servicios

El acceso OAuth está restringido a los [usuarios de prueba](#) que aparecen en la [pantalla de consentimiento de OAuth](#)

ID de cliente

252137327492-8eh8to59s8g7jojvuhot284a8ed8520ts.googleusercontent.com

Secreto del cliente

GOCSPX-1n_SQhpF18mMUumgqzzkaSep39L

Fecha de creación

6 de octubre de 2023, 14:27:50 GMT-1

Estado

Habilitada

DESCARGAR JSON

Librería

pip install social-auth-app-django

Revisar:

<https://dev.to/earthcomfy/django-social-authentication-imk>

Configuración settings.py

```
22 INSTALLED_APPS = [  
23     'django.contrib.admin',  
24     'django.contrib.auth',  
25     'social_django',  
26     'django.contrib.contenttypes',  
27     'django.contrib.sessions',  
28     'django.contrib.messages',  
29     'django.contrib.staticfiles',  
30     'django_cleanup',  
31     'django_ckeditor_5',  
32     'core',  
33     'blog',  
34     'pages',  
35     'services',  
36     'contact',  
37 ]
```

```
38  
39 AUTHENTICATION_BACKENDS = (  
40     'social_core.backends.google.GoogleOAuth2',  
41     'django.contrib.auth.backends.ModelBackend',  
42 )
```

```
AUTHENTICATION_BACKENDS = (  
    'social_core.backends.google.GoogleOAuth2',  
    'django.contrib.auth.backends.ModelBackend',  
)
```

```
56 TEMPLATES = [  
57     {  
58         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
59         'DIRS': [],  
60         'APP_DIRS': True,  
61         'OPTIONS': {  
62             'context_processors': [  
63                 'django.template.context_processors.debug',  
64                 'django.template.context_processors.request',  
65                 'django.contrib.auth.context_processors.auth',  
66                 'django.contrib.messages.context_processors.messages',  
67                 # Add the following two  
68                 'social_django.context_processors.backends',  
69                 'social_django.context_processors.login_redirect',  
70             ],  
71         },  
72     },  
73 ]
```

```
'social_django.context_processors.backends',  
'social_django.context_processors.login_redirect',
```

```

231 | SOCIAL_AUTH_URL_NAMESPACE = 'social'
232 | SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = '10905936
233 | SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = 'GOCSP
234
235 | LOGIN_REDIRECT_URL = 'core:home'
236 | LOGOUT_REDIRECT_URL = 'core:home'

```

```

SOCIAL_AUTH_URL_NAMESPACE = 'social'
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = 'XXXX'
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = 'XXXX'

```

```

LOGIN_REDIRECT_URL = 'core:home'
LOGOUT_REDIRECT_URL = 'core:home'

```

Urls.py

```

12 | urlpatterns = [
13 |     path('admin/', admin.site.urls),
14 |     path('', include(core_urlpatterns)),
15 |     path('blog/', include(blog_urlpatterns)),
16 |     path('pages/', include(pages_urlpatterns)),
17 |     path('services/', include(services_urlpatterns)),
18 |     path('contact/', include(contact_urlpatterns)),
19 |     re_path(r'^oauth/', include('social_django.urls', namespace='social')),
20 | ]

```

```
re_path(r'^oauth/', include('social_django.urls', namespace='social')),
```

Urls de Core

```

2 | from core import views as views_core
3 | from django.contrib.auth import views as auth_views
4
5 | core_urlpatterns = ([
6 |     path('', views_core.home, name='home'),
7 |     path('about/', views_core.about, name='about'),
8 |     path('visit-us/', views_core.store, name='visit-us'),
9 |     path('logout/', auth_views.LoginView.as_view(), name='logout'),
10 | ], 'core')

```

```
path('logout/', auth_views.LoginView.as_view(), name='logout'),
```

Base.html

```

</li>
<li class="nav-item px-lg-4">
  {% if request.user.is_authenticated %}
  <span class="nav-link text-uppercase text-expanded">{{request.user.get_full_name}}</span>
  <form id='logoutForm' method="post" action="{% url 'core:logout' %}">
    {% csrf_token %}
    <a class="nav-link text-uppercase text-expanded" href="JavaScript: logoutForm.submit();">Salir</a>
  </form>
  {% else %}
  <a class="nav-link text-uppercase text-expanded" href="{% url 'social:begin' 'google-oauth2' %}">Entrar con Google</a>
  {% endif %}
</li>
</ul>

```




```

<li class="nav-item px-lg-4">
{% if request.user.is_authenticated %}
<span class="nav-link text-uppercase text-expanded">{{request.user.get_full_name}}</span>
<form id='logoutForm' method="post" action="{% url 'core:logout' %}">
{% csrf_token %}
<a class="nav-link text-uppercase text-expanded" href="JavaScript: logoutForm.submit();">Salir</a>
</form>
{% else %}
<a class="nav-link text-uppercase text-expanded" href="{% url 'social:begin' 'google-oauth2' %}">Google</a>
{% endif %}
</li>

```

Verificación

Accede al administrador y verifica la información almacenada del usuario de google

PYTHON SOCIAL AUTH		
Associations	+ Añadir	 Modificar
Nonces	+ Añadir	 Modificar
User social auths	+ Añadir	 Modificar