

GOOGLE SIGNIN

IDEO LAB

2025

Phase 1 : Configuration dans la Google Cloud Console

Avant de toucher au code, vous devez "déclarer" votre site web chez Google pour obtenir une clé d'identification.

1. Créer un projet :

- Allez sur la [Google Cloud Console](#).
- Créez un **Nouveau Projet** et nommez-le (ex: "Ideo-Lab Login").

2. Configurer l'écran de consentement OAuth :

- Dans le menu de gauche, allez sur **API et services** > **Écran de consentement OAuth**.
- Choisissez **Externe** (puisque vos utilisateurs sont des visiteurs publics).
- Remplissez les infos de base (Nom de l'application : "Ideo-Lab", email support, logo).

3. Créer les identifiants (Le plus important) :

- Allez dans **Identifiants** (Credentials).
- Cliquez sur **+ CRÉER DES IDENTIFIANTS** > **ID client OAuth**.
- **Type d'application** : Application Web.
- **Origines JavaScript autorisées** : C'est ici qu'il faut être précis. Ajoutez l'URL exacte de votre site :
 - `https://www.ideo-lab.com`
 - `https://ideo-lab.com`
- **URI de redirection autorisés** : Si vous utilisez le mode popup (le plus simple), vous

3. Créer les identifiants (Le plus important) :

- Allez dans **Identifiants** (Credentials).
- Cliquez sur + **CRÉER DES IDENTIFIANTS** > **ID client OAuth**.
- **Type d'application** : Application Web.
- **Origines JavaScript autorisées** : C'est ici qu'il faut être précis. Ajoutez l'URL exacte de votre site :
 - `https://www.ideo-lab.com`
 - `https://ideo-lab.com`
- **URI de redirection autorisés** : Si vous utilisez le mode popup (le plus simple), vous n'avez souvent rien à mettre ici, ou bien l'URL de votre page de connexion.
- Cliquez sur **Créer**.

Notez précieusement votre "ID Client" (il ressemble à

`123456...apps.googleusercontent.com`). Vous en aurez besoin pour le code.

GOOGLE URL

- <https://console.cloud.google.com/auth/clients?authuser=3&project=i-deo-lab-login>



Vous travaillez dans [Ideo-lab Login](#) .

Numéro du projet : 972885369398  ID du projet : ideo-lab-login 

[Tableau de bord](#) [Cloud Hub](#) [Nouveau](#)


[+ Créer une VM](#)

[+ Exécuter une requête dans BigQuery](#)


[+ Déployer une application](#)


[+ Créer un bucket de stockage](#)


Rejoignez le Google Developer Program pour acquérir de nouvelles compétences et découvrir des outils de développement sans encourir de frais.


[Rejoindre](#) 


Accès rapide

 API et services


 IAM et administration

 Facturation

 Compute Engine

 Cloud Storage

 BigQuery

 Réseau VPC

 Kubernetes Engine

[Afficher tous les produits](#)



Vous travaillez dans Ideo-lab Login .

Numéro du projet : 972885369398 ID du projet : ideo-lab-login

[Tableau de bord](#)

[Cloud Hub](#)

[Nouveau](#)

The screenshot shows the Google Cloud console interface. At the top, there's a search bar and navigation tabs for 'Tableau de bord', 'Activité', and 'Recommandations'. The main content area is divided into several sections:

- Informations sur le projet:** Displays project details such as 'Nom du projet: Ideo-lab Login', 'Numéro du projet: 972885369398', and 'ID du projet: ideo-lab-login'. It includes a link to 'Ajouter des utilisateurs à ce projet' and 'Accéder aux paramètres du projet'.
- Ressources:** Lists various Google Cloud services pinned to the dashboard, including BigQuery, SQL, Compute Engine, Stockage, Fonctions Cloud Run, and Cloud Run.
- API API:** A section showing API usage metrics. It features a line chart with the title 'Requêtes (requêtes/s)'. The chart shows a single data point at 1 request per second. A warning message states: 'Aucune donnée disponible pour la période sélectionnée.' Below the chart is a link to 'Accéder à l'aperçu des API'.

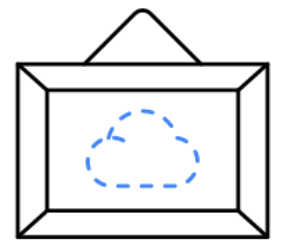
On the left side, there's a sidebar menu with options like 'Cloud Hub', 'Présentation Cloud', 'Solutions', and 'Consultations récentes'. Below that, there's a 'Produits épinglés' section with a list of services like 'Facturation', 'IAM et administrati...', 'Marketplace', 'API et services', 'Vertex AI', 'Compute Engine', 'Kubernetes Engine', 'Cloud Storage', 'Sécurité', 'BigQuery', and 'Monitoring'. A button at the bottom of the sidebar says 'Afficher tous les produits'.

- RPI API et services
- Vertex AI
- Compute Engine
- Kubernetes Engine
- Cloud Storage

- API et services activés
- Bibliothèque
- Identifiants
- Écran de consentement OAuth
- Page des accords d'utilisation



Présentation d'OAuth



Google Auth Platform pas encore configuré
Commencez à configurer l'identité de votre application, et gérez les identifiants pour les appels d'API Google et Se connecter avec Google. [En savoir plus](#)

Premiers pas

Client OAuth créé

Vous pouvez toujours accéder à l'ID client depuis l'onglet "Clients" de Google Auth Platform.



L'accès OAuth est réservé aux [utilisateurs de test](#) listés sur votre [écran de consentement OAuth](#)

ID client 972885369398-3460peb23gvej6fb923l477jb0gfk9.apps.googleusercontent.com



À partir de juin 2025, vous ne pourrez plus afficher ni télécharger le code secret du client une fois que vous aurez fermé cette boîte de dialogue. Assurez-vous d'avoir copié ou téléchargé les informations ci-dessous et de les avoir stockées en lieu sûr.

Code secret du client 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

Date de création 25 novembre 2025, 22:23:27 GMT+1

État Activé

Télécharger au format JSON

OK

Présentation

Branding

Audience

Clients

Accès aux données

Centre de validation

Paramètres

Clients

+ Créer un client

Supprimer

Restaurer les clients OAuth supprimés

ID clients OAuth 2.0

<input type="checkbox"/>	Nom	Date de création ↓	Type	ID client	Actions
<input type="checkbox"/>	ideolab	25 nov. 2025	Application Web	972885369398-3460...	

Phase 2 – Intégration dans le site web

Phase 2 : Intégration sur votre site Web

Google propose une nouvelle librairie (Google Identity Services) beaucoup plus simple qu'avant. Voici comment l'ajouter à votre code HTML.

1. Ajoutez le script Google Placez ceci dans le `<head>` ou juste avant la fermeture du `<body>` de votre page de connexion :

HTML



```
<script src="https://accounts.google.com/gsi/client" async defer></script>
```

2. Ajoutez le bouton (Code HTML) À l'endroit où vous voulez que le bouton apparaisse, insérez ce code. Remplacez `VOTRE_CLIENT_ID` par celui obtenu en Phase 1.

HTML

C

```
<div id="g_id_onload"
  data-client_id="VOTRE_CLIENT_ID"
  data-context="signin"
  data-ux_mode="popup"
  data-callback="handleCredentialResponse"
  data-auto_prompt="false">
</div>

<div class="g_id_signin"
  data-type="standard"
  data-shape="rectangular"
  data-theme="outline"
  data-text="signup_with"
  data-size="large"
  data-logo_alignment="left">
</div>
```

3. Gérez la réponse (JavaScript) Lorsque l'utilisateur se connecte, Google renvoie un "token" (jeton) chiffré. Vous devez le récupérer avec une fonction JavaScript. Ajoutez ceci dans une balise

`<script>` :

JavaScript

C

```
function handleCredentialResponse(response) {  
  // Le token d'identification est dans response.credential  
  console.log("Token Google ID encoded: " + response.credential);  
  
  // ICI : Vous devez envoyer ce token à votre serveur (backend)  
  // pour créer la session utilisateur ou l'inscrire dans votre base de données.  
}
```

Phase 3 : La validation côté Serveur (Backend)

Attention : Ne faites jamais confiance aveuglément aux données reçues côté navigateur.

1. Le JavaScript envoie le `response.credential` (qui est un JWT - JSON Web Token) à votre serveur (via une requête POST Ajax/Fetch).
2. Votre serveur (PHP, Node.js, Python, etc.) reçoit ce token.
3. Votre serveur utilise une librairie Google pour **vérifier** que le token est valide et qu'il vient bien de Google.
4. Une fois validé, vous récupérez l'email (`email@gmail.com`) et le nom depuis le token, et vous créez l'utilisateur dans votre base de données `ideo-lab` .

Étape 1 : Installer la librairie Google

Votre serveur Django a besoin d'un outil pour "décoder" la preuve d'identité envoyée par Google.

Dans votre terminal, installez ce paquet :

```
Bash
```

```
C
```

```
pip install google-auth
```

Python

C

```
import json
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.contrib.auth import login, get_user_model
from django.contrib.auth.models import User
from google.oauth2 import id_token
from google.auth.transport import requests as google_requests

# VOTRE CLIENT ID (Celui qui est dans votre HTML)
GOOGLE_CLIENT_ID = "972885369398-3460peb23gvejuj6fb923l477jb0gfk9.apps.googleusercontent.com"

@csrf_exempt
def google_auth_receiver(request):
    """
    Reçoit le token Google depuis le Javascript, le vérifie,
    et connecte/crée l'utilisateur Django correspondant.
    """
    if request.method == "POST":
        try:
            # 1. On récupère les données envoyées par le JS
            data = json.loads(request.body)
            token = data.get('credential')
```

```
# 3. On récupère les infos utiles
email = idinfo['email']
first_name = idinfo.get('given_name', '')
last_name = idinfo.get('family_name', '')

# 4. On cherche l'utilisateur ou on le crée
User = get_user_model()

try:
    # Cas A : L'utilisateur existe déjà
    user = User.objects.get(email=email)
except User.DoesNotExist:
    # Cas B : C'est un nouvel utilisateur, on le crée
    # On utilise l'email comme username pour éviter les doublons
    user = User.objects.create(
        username=email,
        email=email,
        first_name=first_name,
        last_name=last_name
    )
    user.set_unusable_password() # Pas de mot de passe car géré par Google
    user.save()

# 5. On vérifie si le compte est actif
if not user.is_active:
    return JsonResponse({'success': False, 'message': 'Compte désactivé'})

# 6. On connecte l'utilisateur (Création de la session Django)
login(request, user)
```

DJANGO SETTINGS

Python

C

```
# settings.py
```

```
# Important pour que la popup Google puisse "parler" à votre fenêtre principale  
SECURE_CROSS_ORIGIN_OPENER_POLICY = 'same-origin-allow-popups'
```

Dans votre fichier `base.html` (ou dans le `<head>` de `login.html`), ajoutez cette ligne tout en haut :

HTML

C

```
<meta name="referrer" content="no-referrer-when-downgrade">
```