

Este es un **README.md** estructurado para que cualquier desarrollador de tu equipo (o tú mismo en el futuro) pueda replicar la configuración sin errores y, sobre todo, sin comprometer la seguridad.

Configuración de Google Drive API & Autenticación

Este documento detalla el flujo para configurar una **Service Account**, generar credenciales y asegurar el acceso a carpetas en **Shared Drives** (Unidades Compartidas) desde Django.

1. Creación del Proyecto y Service Account

1. Ve a [Google Cloud Console](#).
2. Crea un nuevo proyecto o selecciona uno existente.
3. En el menú lateral, ve a **APIs y Servicios > Biblioteca**.
4. Busca "**Google Drive API**" y haz clic en **Habilitar**.
5. Ve a **APIs y Servicios > Credenciales**.
6. Haz clic en **Crear credenciales > Cuenta de servicio**.
7. Asigna un nombre (ej. django-drive-manager) y finaliza la creación.

2. Generación del Archivo de Claves (JSON)

⚠ IMPORTANTE: Nunca subas este archivo a GitHub. Google inhabilitará la clave automáticamente si se detecta en un repositorio público.

1. Dentro de la sección **Cuentas de servicio**, haz clic en el email de la cuenta creada.
2. Ve a la pestaña **Claves (Keys)**.
3. Haz clic en **Agregar clave > Crear clave nueva**.
4. Selecciona el formato **JSON** y descárgalo.
5. Renombra el archivo a algo identificable (ej. drive-credentials.json) y guárdalo en la raíz de tu proyecto (asegúrate de que esté en tu .gitignore).

3. Permisos en Google Drive (Paso Crítico)

Para que la Service Account pueda leer y **eliminar** archivos, debe ser miembro de la carpeta o unidad:

1. Copia el **Email de la Service Account** (ej. nombre@proyecto.iam.gserviceaccount.com).
2. Ve a tu Google Drive y localiza la carpeta o **Unidad Compartida**.
3. Haz clic derecho > **Compartir** (o Gestionar miembros).

4. Pega el email de la cuenta de servicio.
5. **Asigna el Rol:**
 - **Gestor de contenido:** Necesario para poder eliminar archivos.
 - **Colaborador:** Solo permite eliminar lo que la cuenta misma subió.

4. Implementación en Django

Carga de Credenciales

Utilizamos la librería google-auth y google-api-python-client.

Python

```
from google.oauth2 import service_account
from googleapiclient.discovery import build

def get_drive_service():
    SCOPES = ['https://www.googleapis.com/auth/drive']
    # Carga desde archivo local (asegurado por .gitignore)
    creds = service_account.Credentials.from_service_account_file(
        'drive-credentials.json', scopes=SCOPES
    )
    return build('drive', 'v3', credentials=creds)
```

Flujo de Eliminación Segura

Para borrar archivos en Unidades Compartidas, es obligatorio el parámetro supportsAllDrives.

Python

```
def delete_file(file_id):
    service = get_drive_service()
    service.files().delete(
        fileId=file_id,
        supportsAllDrives=True # Requerido para Shared Drives
    ).execute()
```



5. Resolución de Errores Comunes

Error	Causa Probable	Solución
invalid_grant: Invalid JWT Signature	Clave expirada o inhabilitada por Google (leak en GitHub).	Generar una clave nueva en la consola y borrar la vieja.
404 File Not Found (al borrar)	Falta de permisos de escritura/borrado.	Cambiar el rol de la Service Account a Gestor de contenido .
insufficientPermissions	El Scope definido en Python es muy limitado.	Asegurarse de usar el scope .../auth/drive.

Siguiente paso recomendado: ¿Te gustaría que añadamos una sección sobre cómo rotar estas claves automáticamente cada 90 días para cumplir con estándares de seguridad?