

Para tu caso de uso donde múltiples usuarios de tu aplicación Django necesitan subir archivos a un único Google Drive, te recomiendo las siguientes alternativas:

⌚ Alternativa Recomendada: Google Drive API con Service Account

Esta es la mejor opción para tu caso porque:

- **Un único punto de autenticación:** La aplicación se autentica con una Service Account, no cada usuario individual
- **Todos los archivos van al mismo Drive:** Compartes una carpeta del Drive con la Service Account
- **No requiere interacción del usuario:** La subida es transparente

Implementación paso a paso:

1. Crear Service Account en Google Cloud Console:

1. Ve a console.cloud.google.com
2. Crea un proyecto o selecciona uno existente
3. Habilita la "Google Drive API"
4. Ve a "Credentials" → "Create Credentials" → "Service Account"
5. Descarga el archivo JSON de credenciales

2. Compartir la carpeta de Drive:

1. Crea una carpeta en Google Drive donde llegarán los archivos
2. Comparte esa carpeta con el email de la Service Account
(algo como: tu-service@proyecto.iam.gserviceaccount.com)
3. Dale permisos de "Editor"

3. Instalar dependencias:

```
pip install google-api-python-client google-auth google-auth-httplib2
```

4. Código de ejemplo para Django:

```
# services/google_drive_service.py
from google.oauth2 import service_account
from googleapiclient.discovery import build
from googleapiclient.http import MediaFileUpload
import os

class GoogleDriveService:
    SCOPES = ['https://www.googleapis.com/auth/drive.file']
```

```

def __init__(self):
    # Ruta al archivo de credenciales JSON
    credentials_path = os.environ.get('GOOGLE_CREDENTIALS_PATH')
    self.folder_id = os.environ.get('GOOGLE_DRIVE_FOLDER_ID')

    credentials = service_account.Credentials.from_service_account_file(
        credentials_path,
        scopes=self.SCOPES
    )
    self.service = build('drive', 'v3', credentials=credentials)

    def upload_file(self, file_path: str, file_name: str, mime_type: str =
'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'):
        """
        Sube un archivo al Google Drive compartido

        Args:
            file_path: Ruta local del archivo a subir
            file_name: Nombre con el que se guardará en Drive
            mime_type: Tipo MIME del archivo (por defecto Excel)

        Returns:
            dict con id y webViewLink del archivo subido
        """
        file_metadata = {
            'name': file_name,
            'parents': [self.folder_id]
        }

        media = MediaFileUpload(
            file_path,
            mimetype=mime_type,
            resumable=True
        )

        file = self.service.files().create(
            body=file_metadata,
            media_body=media,
            fields='id, webViewLink'
        ).execute()

```

```
    return file
```

5. Uso en tu vista Django:

```
# views.py
from django.http import JsonResponse
from .services.google_drive_service import GoogleDriveService
import os
from datetime import datetime

def submit_inspection(request):
    if request.method == 'POST':
        # ... tu lógica actual para generar el Excel ...

        # Generar nombre único con timestamp y usuario
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        user = request.user.username
        file_name = f"inspeccion_{user}_{timestamp}.xlsx"

        # Ruta donde se guarda temporalmente el Excel
        temp_file_path = f"/tmp/{file_name}"

        # Generar el Excel (tu código existente)
        generate_excel_from_template(request.POST, temp_file_path)

        # Subir a Google Drive
        try:
            drive_service = GoogleDriveService()
            result = drive_service.upload_file(temp_file_path, file_name)

            # Opcional: eliminar archivo temporal
            os.remove(temp_file_path)

        return JsonResponse({
            'success': True,
            'message': 'Inspección enviada correctamente',
            'drive_link': result.get('webViewLink')
```

```
        })
    except Exception as e:
        return JsonResponse({
            'success': False,
            'error': str(e)
        }, status=500)
```

6. Variables de entorno necesarias:

```
# .env
GOOGLE_CREDENTIALS_PATH=/path/to/your/service-account-credentials.json
GOOGLE_DRIVE_FOLDER_ID=1ABC123xyz... # ID de la carpeta compartida
```