

```

import requests
import webbrowser
from urllib.parse import urlencode, urlparse, parse_qs

# Configurações do OAuth2
client_id = "SEU_CLIENT_ID"
client_secret = "SEU_CLIENT_SECRET"
redirect_uri = "http://localhost"
auth_url = "https://accounts.google.com/o/oauth2/v2/auth"
token_url = "https://oauth2.googleapis.com/token"
scope = "https://www.googleapis.com/auth/userinfo.profile"

# Passo 1: Gerar a URL de autorização
params = {
    "client_id": client_id,
    "redirect_uri": redirect_uri,
    "response_type": "code",
    "scope": scope,
    "access_type": "offline", # Permite a geração de refresh tokens
    "prompt": "consent",      # Garante que o consentimento será solicitado
}

auth_request_url = f"{auth_url}?{urlencode(params)}"

# Passo 2: Abrir a URL no navegador
print("Abrindo o navegador para autenticação...")
webbrowser.open(auth_request_url)

# Passo 3: O usuário fornece o código retornado após autenticação
authorization_response = input("Cole aqui a URL completa recebida após a autenticação: ")

# Extrair o código de autorização da URL
parsed_url = urlparse(authorization_response)
authorization_code = parse_qs(parsed_url.query).get("code")[0]

# Passo 4: Trocar o código de autorização por um token de acesso
token_data = {
    "client_id": client_id,
    "client_secret": client_secret,
    "code": authorization_code,
    "redirect_uri": redirect_uri,
    "grant_type": "authorization_code",
}
token_response = requests.post(token_url, data=token_data)

if token_response.status_code == 200:
    tokens = token_response.json()
    print("Autorização bem-sucedida!")
    print("Access Token:", tokens.get("access_token"))
    print("Refresh Token:", tokens.get("refresh_token"))
else:
    print("Erro ao obter o token:", token_response.json())

```