Documento Técnico: Integración de Login con Google en una Página Web

(Versión 2.0 - Julio 2024)

---

# 1. Configuración del Proyecto en Google Cloud

Objetivo: Generar las credenciales necesarias para autenticar usuarios.

* \*\*Crear un proyecto:\*\*  
  - Accede a https://console.cloud.google.com/.  
  - Haz clic en “Crear Proyecto” y asigna un nombre (ej: "Login-Google-MiWeb").
* \*\*Configurar la pantalla de consentimiento:\*\*  
  - Menú lateral → APIs y Servicios → Pantalla de Consentimiento OAuth.  
  - Tipo de usuario: Selecciona \*Externo\*.  
  - Información básica:  
   - Nombre de la aplicación: Ej: "Mi Web".  
   - Correo de soporte: Ingresa tu email.  
  - Scopes (Permisos):  
   - Añade manualmente:  
   - https://www.googleapis.com/auth/userinfo.email  
   - https://www.googleapis.com/auth/userinfo.profile  
  - Completa los pasos hasta “Guardar y Continuar”.
* \*\*Generar credenciales OAuth 2.0:\*\*  
  - Menú lateral → APIs y Servicios → Credenciales.  
  - Haz clic en + Crear Credenciales → ID de Cliente OAuth.  
  - Tipo de aplicación: Aplicación Web.  
  - URI de redireccionamiento autorizados:  
   - Desarrollo: http://localhost:3000/auth/google/callback  
   - Producción: https://tudominio.com/auth/google/callback  
  - Client ID: Copia el valor generado (ej: 1234-xxxx.apps.googleusercontent.com).

# 2. Implementación en el Frontend

Objetivo: Mostrar el botón de Google y manejar la respuesta de autenticación.

## Código HTML

<!-- Botón personalizado -->  
<button id="btn-google-login" class="custom-google-btn">  
 <img src="ruta/al/icono-google.png" alt="Google Icon">  
 Continuar con Google  
</button>  
  
<!-- Cargar biblioteca de Google -->  
<script src="https://accounts.google.com/gsi/client"></script>

## JavaScript

document.getElementById("btn-google-login").addEventListener("click", () => {  
 google.accounts.id.initialize({  
 client\_id: "TU\_CLIENT\_ID.apps.googleusercontent.com",  
 callback: (response) => {  
 const token = response.credential;  
 fetch("/auth/google", {  
 method: "POST",  
 headers: { "Content-Type": "application/json" },  
 body: JSON.stringify({ token })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) window.location.href = "/dashboard";  
 else alert("Error: " + data.error);  
 });  
 }  
 });  
 google.accounts.id.prompt();  
});

# 3. Backend: Verificación del Token (Node.js/Express)

Objetivo: Validar el token recibido y autenticar al usuario.

## Dependencias requeridas

npm install express google-auth-library

## Código del Endpoint

const express = require("express");  
const { OAuth2Client } = require("google-auth-library");  
const app = express();  
const client = new OAuth2Client("TU\_CLIENT\_ID");  
  
app.use(express.json());  
  
app.post("/auth/google", async (req, res) => {  
 try {  
 const { token } = req.body;  
  
 const ticket = await client.verifyIdToken({  
 idToken: token,  
 audience: "TU\_CLIENT\_ID"  
 });  
  
 const payload = ticket.getPayload();  
 const userData = {  
 id: payload.sub,  
 email: payload.email,  
 name: payload.name,  
 picture: payload.picture  
 };  
  
 res.json({ success: true, user: userData });  
  
 } catch (error) {  
 console.error("Error en autenticación:", error);  
 res.status(401).json({ success: false, error: "Token inválido" });  
 }  
});  
  
app.listen(3000, () => console.log("Servidor en puerto 3000"));

# 4. Consideraciones Clave

* \*\*Seguridad:\*\*  
  - HTTPS Obligatorio: En producción, usa certificados SSL.  
  - Validar el Audience: debe coincidir con tu Client ID.
* \*\*Errores Comunes:\*\*  
  - Error 401 (Token Inválido): Token expirado o Client ID incorrecto.  
  - Error 400 (Redirect URI Mismatch): Las URIs no coinciden con las configuradas en Google Cloud.

# 5. Personalización y Mejoras

* - Diseño del Botón: Usa CSS personalizado.
* - Manejo de Sesiones: Usa JWT o cookies seguras.
* - Registro en Base de Datos: Guarda `userData.id` como ID único.

# Recursos Adicionales

* Guía de Google Identity Services: https://developers.google.com/identity/gsi/web
* OAuth2 en Node.js: https://github.com/googleapis/google-auth-library-nodejs
* Soporte Técnico: soporte@midominio.com

---

\*Documento creado por [Tu Nombre o Equipo] - © 2024\*

\*Última revisión: 15/07/2024\*

\*Nota: Este documento está optimizado para copiar y pegar en Word. Usa la función "Pegar como texto sin formato" y ajusta los estilos según tus necesidades.\*