Complete Step-by-Step Guide: Setting Up Google Calendar Sync for Your WordPress Plugin

This guide walks you through creating a Google OAuth app from scratch using Google Cloud Platform (GCP).

The goal is to get two key values: Client ID and Client Secret, which your plugin needs to sync appointments one-way from WordPress to Google Calendar. By following these steps, you'll set the app to "In production" mode, ensuring the authorization tokens are long-lived (they won't expire every 7 days, unlike in testing mode). **This prevents frequent reconnections.Important Prerequisites:**

- A Google account (free Gmail or Google Workspace). If you don't have one, create it at accounts.google.com.
- Your WordPress site must use HTTPS (secure domain, e.g., https://your-site.com). Google requires this for redirect URIs.
- Estimated time: 20-45 minutes (longer if you're new to Google Cloud).
- Cost: Completely free—no credit card needed.
- Browser recommendation: Use Google Chrome for the best experience.
- Scope details: The plugin uses "sensitive" permissions (e.g., creating/editing events in Google Calendar). This means you'll see a "This app isn't verified" warning during authorization—it's normal and safe for trusted apps like this. We'll explain how to bypass it.
- Official Google reference: Google Cloud OAuth Setup Docs.

If you encounter errors, check the "Common Issues & Fixes" section at the end.

Step 1: Access Google Cloud Console and Create a New ProjectA "project" is like a container for your app's settings in Google Cloud.

1. Open your web browser and go to console.cloud.google.com.
2. Sign in with your Google account (the same one you'll use for Google Calendar).
3. At the top-left, click the dropdown menu labeled "Select a project" (it might say "No organization" if you're new).
4. Click New Project in the popup window.
5. Enter a project name (e.g., "WP-Appointment-Sync").
    - Tip: Keep it descriptive but simple—no spaces or special characters.
6. Leave "Location" as default (or select your organization if using Google Workspace).
7. Click Create. Wait 10-30 seconds for it to process.
8. Once created, select your new project from the top dropdown menu.Screenshot description: The top bar shows a search box and dropdown; the "New Project" button is blue in the popup.

Common Pitfall: If you see a billing prompt, skip it—OAuth setup doesn't require billing.

Step 2: Enable the Google Calendar APIThis activates the API your plugin needs to interact with Calendar.

1. In the left sidebar, click APIs & Services > Library.
2. In the search bar, type "Google Calendar API".
3. Click the result titled Google Calendar API.
4. Click the blue Enable button.
5. Wait a few seconds for activation (you'll see a success message).Screenshot description: The Library page has a search bar at the top; the API card shows a calendar icon.

Why this step? Without enabling the API, your app can't access Calendar data. Reference: Enable APIs.

Step 3: Configure the OAuth Consent ScreenThis screen defines what users see when authorizing your app (e.g., "Allow this app to access your Calendar?").

1. In the left sidebar, click APIs & Services > OAuth consent screen.
2. If not set up, click Configure consent screen.
3. Under "User Type", select External (this allows non-Google Workspace users; most people choose this).
4. Click Create.
5. App information section:
    - App name: Enter something user-friendly (e.g., "WordPress Appointment Sync").
    - User support email: Your Gmail address (for user questions).
    - App logo: Optional—upload a 120x120px image if you have one (e.g., your site logo). Skip if not.
    - App domain: Enter your website's homepage (e.g., https://your-site.com).
    - Authorized domains: Add your site's domain (e.g., your-site.com) without "https://".
    - Developer contact information: Your Gmail again.
    - Click Save and continue.
6. Scopes section:
    - Click Add or remove scopes.
    - In the search bar, type "calendar" and check these scopes (permissions):

        - https://www.googleapis.com/auth/calendar.events (required for creating/editing events).
        - Optionally: https://www.googleapis.com/auth/calendar (full calendar access if your plugin needs it—check plugin docs).
    - Click Update > Save and continue.
7. Test users section: Skip adding any emails (we'll publish to production soon). Click Save and continue.
8. Summary section: Review everything, then click Back to dashboard.Screenshot description: The consent screen has tabs like "App information" and "Scopes"; scopes list shows checkboxes.

Tip: Be accurate here—Google uses this for the authorization popup. If you make changes later, users might need to reauthorize.

Step 4: Create OAuth Credentials (Client ID and Client Secret)These are the two values your plugin needs.

1. In the left sidebar, click APIs & Services > Credentials.
2. At the top, click Create Credentials > OAuth client ID.
3. Under "Application type", select Web application.
4. Name: Enter a name (e.g., "WP Sync Client").
5. Authorized redirect URIs: This is crucial—it's where Google sends users after authorization.
    - Copy the exact URI from your plugin's settings page (e.g., go to WordPress Admin > Plugin Settings > Google Calendar section—it often shows or generates something like (`https://sitename.com/wp-admin/admin.php?page=wpappsws-google-calendar`).
    - Paste it here. Add multiple if needed (e.g., for staging sites).
6. Click Create.
7. A popup shows your credentials—copy and save them securely:
    - Client ID: Looks like 1234567890-abcde.apps.googleusercontent.com.
    - Client Secret: Looks like GOCSPX-abcdef123456.
    - Download the JSON file if prompted (optional backup).

Screenshot description: The Credentials page has a "+ Create Credentials" button; the form has fields for name and URIs.

Security Note: Never share these publicly. If compromised, revoke and recreate them.

Step 5: Publish the App to "In Production" (For Long-Lived Tokens)This removes the 7-day token expiration limit.

1. Go back to APIs & Services > OAuth consent screen.
2. At the top, under "Publishing status", it should say Testing.
3. Click Publish app.
4. In the confirmation popup, ignore warnings about verification (it's not required for limited/personal use). Click Confirm or Publish.
5. Wait 1-5 minutes for changes to propagate. Refresh the page—status should now say In production.Screenshot description: The consent screen top section has a "Publish app" button next to the status.

What this does: In "Testing", tokens expire weekly. In "Production", they're long-lived. Reference: OAuth Publishing Status.

Step 6: Connect the Plugin in WordPress

1. Log in to your WordPress admin dashboard.
2. Go to the plugin's settings page (e.g., Plugins > [Your Plugin Name] > Settings).
3. Enter the Client ID and Client Secret from Step 4.
4. Save changes.
5. Click the "Connect to Google" or "Authorize" button.
6. A Google popup appears:
    - Sign in with your Google account (the one with your Calendar).
    - You'll see: "Google hasn't verified this app" warning (normal for non-verified apps).
    - Click Advanced (small link at the bottom).
    - Click Go to [App Name] (unsafe) or Continue.
    - Review scopes and click Continue or Allow.
7. The popup closes, and the plugin should show "Connected successfully".
8. Test: Create an appointment in WordPress—it should sync to your Google Calendar.

Step 7: Verify Long-Lived Tokens (Optional but Recommended Test)

1. Optionally revoke current access: Go to myaccount.google.com/permissions > Find your app > Remove access.
2. Reconnect the plugin (repeat Step 6).
3. Wait 8-14 days (use your site normally).
4. After that, create/update an appointment—if it syncs without reauthorization, tokens are long-lived!

Common Issues & Fixes

- "This app isn't verified" warning won't go away? It's expected without full verification (which takes weeks and requires a privacy policy/video). Always click Advanced > Continue—it's safe.
- Redirect URI mismatch error? Double-check the URI in Step 4 matches exactly what the plugin provides (case-sensitive, including query params).
- Token expires after 7 days anyway? Ensure publishing status is "In production" (Step 5). If not, republish.
- Scope errors (e.g., insufficient permissions)? Go back to Step 3 and add missing scopes, then recreate credentials.
- Sync not working? Clear browser cache, try incognito mode, or check plugin logs for errors.
- User cap reached (~100 users)? If you have many sites/users, request verification: Prepare a privacy policy page on your site and submit via the consent screen's "Verification" button.
- Still stuck? Contact support with screenshots of the error.

Congratulations! Your sync is now set up with reliable, long-lasting tokens. If issues persist, share error details for help.