

Google Cloud Setup for Anderson's Library

Project Details

- **Project Name:** anderson-library
- **Project Number:** 906077568035
- **Purpose:** Google Drive authentication and file management

Step 1: Access Your Project

1. Go to [Google Cloud Console](#)
2. **Select project:** anderson-library (906077568035)
3. If not visible, click the project dropdown and search for "anderson-library"

Step 2: Enable Required APIs

Navigate to **APIs & Services** → **Library** and enable:

Required APIs:

- ☒ **Google Drive API**
- ☒ **Google Sheets API**
- ☒ **Google Identity Toolkit API** (for OAuth)

How to enable:

1. Search for each API name
2. Click on it → Click "**Enable**"
3. Wait for confirmation message

Step 3: Create OAuth 2.0 Client ID

1. Go to **APIs & Services** → **Credentials**
2. Click **+ CREATE CREDENTIALS** → **OAuth 2.0 Client ID**
3. Configure:
 - **Application type:** Web application
 - **Name:** Anderson's Library Web Client
 - **Authorized JavaScript origins:**

`http://localhost:8080`
`http://127.0.0.1:8080`
`https://bowersworld.com`
`https://www.bowersworld.com`
`https://callmechewy.github.io`

- **Authorized redirect URIs:**

`http://localhost:8080/library/auth/callback`
`https://bowersworld.com/library/auth/callback`
`https://callmechewy.github.io/BowersWorld-com/library/auth/callback`

4. Click **"Create"**

5. **Copy the Client ID** (format: `123456-abcdef.apps.googleusercontent.com`)

Step 4: Create API Key

1. Still in **APIs & Services** → **Credentials**
2. Click **+ CREATE CREDENTIALS** → **API key**
3. **Copy the API key** (format: `AlzaSyC...`)
4. Click **"Restrict Key"** to secure it:

API Key Restrictions:

Application restrictions:

- Select: **HTTP referrers (web sites)**
- Add referrers:

`http://localhost:8080/*`
`http://127.0.0.1:8080/*`
`https://bowersworld.com/*`
`https://www.bowersworld.com/*`
`https://callmechewy.github.io/*`

API restrictions:

- Select: **Restrict key**
- Select APIs:
 - Google Drive API
 - Google Sheets API

- Google Identity Toolkit API

5. Click "Save"

Step 5: Configure OAuth Consent Screen

1. Go to **APIs & Services** → **OAuth consent screen**
2. **User Type:** External (unless you have Google Workspace)
3. **Fill required fields:**
 - **App name:** Anderson's Library
 - **User support email:** herbbowers@gmail.com
 - **App domain:** bowersworld.com
 - **Developer contact:** herbbowers@gmail.com
4. **Scopes:** Add these scopes:
 - <https://www.googleapis.com/auth/drive.readonly>
 - <https://www.googleapis.com/auth/userinfo.email>
 - <https://www.googleapis.com/auth/spreadsheets>
5. **Test users:** Add your email for testing

Step 6: Update Your Configuration

Copy your credentials into the configuration:

javascript

```
const LIBRARY_CONFIG = {  
  // Your actual credentials from anderson-library project  
  clientId: 'YOUR_CLIENT_ID_HERE.apps.googleusercontent.com',  
  apiKey: 'YOUR_API_KEY_HERE',  
  
  // Google Drive folder structure (get these IDs next)  
  folderIds: {  
    books: 'YOUR_BOOKS_FOLDER_ID',  
    covers: 'YOUR_COVERS_FOLDER_ID',  
    database: 'YOUR_DATABASE_FOLDER_ID',  
    system: 'YOUR_SYSTEM_FOLDER_ID'  
  },  
  
  // Google Sheets for user management (create these next)  
  sheetIds: {  
    users: 'YOUR_USERS_SHEET_ID',  
    activity: 'YOUR_ACTIVITY_LOG_SHEET_ID'  
  }  
};
```

Step 7: Create Google Drive Folder Structure

1. Go to Google Drive
2. Create main folder: "Anderson's Library"
3. Inside that folder, create:
 - 📁 **Books** (organize by category: Programming, Math, Physics, etc.)
 - 📁 **Covers** (book cover images)
 - 📁 **Database** (SQLite files, JSON data)
 - 📁 **System** (config files, logs)
4. Get folder IDs:
 - Open each folder → URL will show the ID
 - Format: `https://drive.google.com/drive/folders/[FOLDER_ID_HERE]`
 - Copy each ID for your config

Step 8: Create Google Sheets for User Management

Create "Users" Sheet:

1. **Create new Google Sheet:** "Anderson Library - Users"

2. **Column headers (Row 1):**

A: Email

B: UserID

C: Role

D: Status

E: Created

F: LastLogin

G: Permissions

H: Notes

3. **Share with your admin email**

4. **Copy the Sheet ID** from URL

Create "Activity Log" Sheet:

1. **Create new Google Sheet:** "Anderson Library - Activity Log"

2. **Column headers (Row 1):**

A: Timestamp

B: Email

C: Action

D: Details

E: IP

3. **Share with your admin email**

4. **Copy the Sheet ID** from URL

Step 9: Test Configuration

1. **Update your HTML file** with real credentials

2. **Start local server:** `python3 -m http.server 8080`

3. **Visit:** `http://localhost:8080/Updates/google_drive_auth_setup.html`





4. **Click "Sign in with Google"**

5. **Check console** for success messages

Expected Results

After successful setup:

-  Google sign-in works without errors

-  Can access Google Drive files
-  User data saves to Google Sheets
-  Activity logging functions
-  Ready for production deployment

Security Notes

- **API Key restrictions** prevent unauthorized use
- **OAuth consent screen** shows professional app info
- **Limited scopes** only request necessary permissions
- **Authorized domains** restrict where auth can be used

Next Steps

Once authentication works:

1. **Upload your 1,219 books** to Drive folders
2. **Process book covers** and upload to Covers folder
3. **Set up user approval workflow**
4. **Deploy to GitHub Pages**