

Spotify Downloader via Web

Planned for custom song wishes for weddings.

- [Setup a Spotify Downloader](#)

Setup a Spotify Downloader

📁 Wedding DJ Request System – Setup Guide

This guide walks you through setting up a complete system where guests can submit Spotify track requests through your custom web page. The system will:

- 📁 Only accept valid Spotify **track** links (including intl links)
- 📁 Prevent spam with **rate-limiting** per browser session
- 📁 Send you **email notifications** for new submissions
- 📁 Log all requests into a **Google Sheet**
- 📁 Automatically **download songs** using `spotDL` on your laptop
- 📁 Require a simple **password** to access the form

1. 📁 Create the Google Sheet

1. Go to sheets.new
2. Rename it to: `Wedding DJ Requests`
3. Set up the first row with column headers:

Timestamp | Song | Guest | Session ID

2. 📁 Enable Google Sheets + Drive API

1. Go to [Google Cloud Console](https://console.cloud.google.com)
2. Create a new project

3. Go to **APIs & Services > Library**
 4. Enable:
 - **Google Sheets API**
 - **Google Drive API**
 5. Go to **APIs & Services > Credentials**
 6. Click "**Create Credentials > Service Account**"
 - Give it a name like `dj-sheet-reader`
 7. Go to the **Keys tab** of that account
 - Create a new JSON key → download it (e.g. `credentials.json`)
 8. Share your Google Sheet with the **email in the JSON file**
-

3. Set Up Google Apps Script

1. In your Google Sheet, click:
Extensions > Apps Script
2. Replace everything with:

```
function doPost(e) {  
  const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Sheet1");  
  const song = e.parameter.song;  
  const guest = e.parameter.guest || "Anonymous";  
  const sessionId = e.parameter.session_id || "no-session";  
  const now = new Date();  
  const cooldownMinutes = 5;  
  
  const lastRow = sheet.getLastRow();  
  const numRowsToCheck = Math.min(50, lastRow - 1);  
  const rows = numRowsToCheck > 0  
    ? sheet.getRange(lastRow - numRowsToCheck + 1, 1, numRowsToCheck, 4).getValues()  
    : [];  
  
  for (let i = rows.length - 1; i >= 0; i--) {  
    const [timestamp, previousSong, previousGuest, previousSessionId] = rows[i];  
    if (previousSessionId && previousSessionId === sessionId) {  
      const lastTime = new Date(timestamp);  
      const diffMinutes = (now - lastTime) / 1000 / 60;  
      if (diffMinutes < cooldownMinutes) {  
        return ContentService.createTextOutput("rate-limit").setMimeType(ContentService.MimeType.TEXT);  
      }  
    }  
  }  
}
```

```
break;
}
}

sheet.appendRow([now, song, guest, sessionId]);

MailApp.sendEmail({
  to: "you@example.com", // replace with your email
  subject: "🎵 Neuer Musikwunsch eingegangen",
  body: `Von: ${guest}\nTrack: ${song}`
});

return ContentService.createTextOutput("success").setMimeType(ContentService.MimeType.TEXT);
}
```

3. Click **Deploy > Manage deployments > New deployment**
 - Select type: **Web App**
 - Execute as: **Me**
 - Who has access: **Anyone**
 - Deploy → copy the URL

4. Create the HTML Wish Page

Create a file called `index.html` and paste this:

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>DJ Musikwunsch</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body {
      font-family: sans-serif;
      background: url('background.jpg') no-repeat center center fixed;
      background-size: cover;
      display: flex;
      justify-content: center;
```

```
    align-items: center;
    min-height: 100vh;
    padding: 20px;
}
.container {
    background: rgba(255, 255, 255, 0.95);
    padding: 24px;
    border-radius: 12px;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
    max-width: 400px;
    width: 100%;
}
input, button {
    width: 100%;
    padding: 10px;
    margin-bottom: 16px;
    border: 1px solid #ccc;
    border-radius: 6px;
    font-size: 16px;
}
button {
    background: #333;
    color: white;
    cursor: pointer;
}
button:hover {
    background: #111;
}
#status {
    text-align: center;
    font-weight: bold;
}
</style>
</head>
<body>
<script>
const correctPassword = "hochzeit2024"; // set your password
const unlockPage = () => {
    const input = prompt("☐ Passwort eingeben:");
    if (input !== correctPassword) {
```

```
    alert("❌ Falsches Passwort.");
    window.location.href = "https://www.google.com";
  }
};
window.onload = unlockPage;
</script>
```

```
<div class="container">
  <h2>🎵 Dein Musikwunsch</h2>
  <form id="request-form">
    <label for="song">Spotify Track-Link:</label>
    <input type="text" id="song" placeholder="https://open.spotify.com/track/..." required>
    <label for="guest">Dein Name (optional):</label>
    <input type="text" id="guest" placeholder="z. B. Lisa, Onkel Joe">
    <button type="submit">Wunsch absenden</button>
  </form>
  <p id="status"></p>
</div>
```

```
<script>
const getSessionId = () => {
  const existing = localStorage.getItem("session_id");
  if (existing) return existing;
  const newId = crypto.randomUUID();
  localStorage.setItem("session_id", newId);
  return newId;
};

const isValidSpotifyTrackUrl = url =>
  /^https:\/\/open\.spotify\.com\/(intl-[a-z]{2}\V)?trackV[a-zA-Z0-9]+(\?.*)?$/\.test(url);

const form = document.getElementById('request-form');
const status = document.getElementById('status');

form.addEventListener('submit', async (e) => {
  e.preventDefault();

  const song = document.getElementById('song').value.trim();
  const guest = document.getElementById('guest').value.trim();
  const sessionId = getSessionId();
```

```

if (!isValidSpotifyTrackUrl(song)) {
  status.textContent = "⚠ Bitte nur gültige Spotify *Track*-Links verwenden.";
  status.style.color = "orange";
  return;
}

const formData = new URLSearchParams();
formData.append("song", song);
formData.append("guest", guest);
formData.append("session_id", sessionId);

try {
  const res = await fetch("YOUR_WEBAPP_URL_HERE", { method: 'POST', body: formData });
  const text = await res.text();
  if (text.includes("success")) {
    status.textContent = "✅ Wunsch eingereicht!";
    status.style.color = "green";
    form.reset();
  } else if (text.includes("rate-limit")) {
    status.textContent = "⏱ Bitte kurz warten, bevor du erneut einen Wunsch sendest.";
    status.style.color = "orange";
  } else {
    status.textContent = "⚠ Fehler: " + text;
    status.style.color = "red";
  }
} catch (err) {
  console.error(err);
  status.textContent = "❌ Fehler beim Senden.";
  status.style.color = "red";
}
});
</script>
</body>
</html>

```

“ Replace `YOUR_WEBAPP_URL_HERE` with your actual Google Apps Script URL.

5. 📄 Python Script to Download Songs

1. Install requirements:

```
pip install gspread oauth2client spotdl
```

2. Save this as `downloader.py`:

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials
import subprocess
import time
import os

SHEET_NAME = "Wedding DJ Requests"
DOWNLOAD_FOLDER = "C:/Users/yanik/Music/DJRequests"
PROCESSED_FILE = "downloaded.txt"
SERVICE_ACCOUNT_FILE = "credentials.json"

scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/drive"]
creds = ServiceAccountCredentials.from_json_keyfile_name(SERVICE_ACCOUNT_FILE, scope)
client = gspread.authorize(creds)
sheet = client.open(SHEET_NAME).sheet1

def load_processed():
    if not os.path.exists(PROCESSED_FILE):
        return set()
    with open(PROCESSED_FILE, "r") as f:
        return set(line.strip() for line in f)

def save_processed(processed):
    with open(PROCESSED_FILE, "w") as f:
        for url in processed:
            f.write(url + "\n")

def download_song(url):
    print(f"📄 Downloading: {url}")
```



```

try:
    subprocess.run(["spotdl", url, "--output", DOWNLOAD_FOLDER], check=True)
    return True
except subprocess.CalledProcessError:
    print(f"❌ Failed to download: {url}")
    return False

def main():
    print("🔄 Monitoring Google Sheet for new requests...")
    processed = load_processed()

    while True:
        try:
            rows = sheet.get_all_records()
            for row in rows:
                url = row.get("Song")
                if url and url not in processed and "open.spotify.com/" in url:
                    success = download_song(url)
                    if success:
                        processed.add(url)
                        save_processed(processed)
        except Exception as e:
            print(f"❌ ERROR: {e}")
            time.sleep(15)

if __name__ == "__main__":
    main()

```

3. Place `credentials.json` in the same folder

4. Run the script:

```
python downloader.py
```

✅ Done!

Your guests can now:

- Open the form with a password
- Submit Spotify track links

- Trigger rate-limited submissions
- And your system downloads the tracks automatically ☐