MAS Submod : Backup to persistent to Google Drive

Dependencies, requirements and procedure

a. Node.js and npm installed (download link : Node.js — Download (nodejs.org) , you can find tutorial on how to install on YouTube, if you don't sure how to)

b. Git clone or download my submod folder, place the folder at anywhere you like (relax, later for the Ren'Py submod part, now for Node.js part).

c. Node modules :

execute " npm install googleapis fs readline util progress " on your cmd (make sure path directory has changed to the folder of the directory you downloaded my submod folder)

1. Go to https://console.cloud.google.com/ Google Cloud Console.

2. Click "NEW PROJECT"

3. Enter Project Name and click CREATE.

4. Click at the project you've created

5. Click APIs and services

6. Click ENABLE APIS AND SERVICES

7. Search Google Drive and click google drive api

Welcome to the API Library

The API Library has documentation, links and a smart search experience.

🔍   Google Drive   ✕

google drive api

google drive activity api
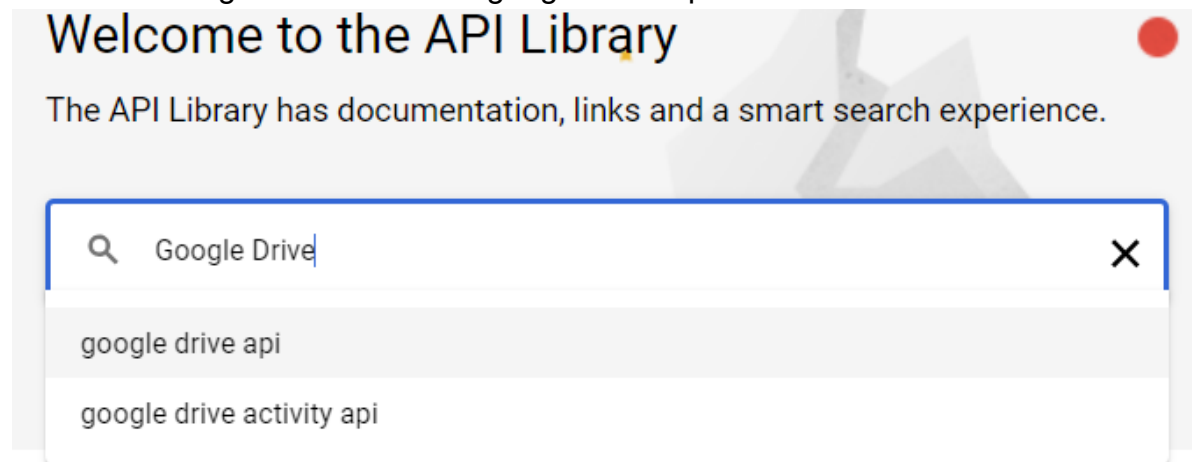
8. Click Google Drive API

11 results

Google Drive API
Google Enterprise API ❓

With the Google Drive API, you can access resources from Google Drive to create files, manage file sharing, search for files and folders, and more.

Google Drive Activity API
Google Enterprise API ❓

With the Google Drive Activity API, you can retrieve information about a user's Google Drive activity. This provides additional functionality on top the Google Drive API for your app to display activity on a user's files, track changes to specific files or folders, or alert a user to new comments or changes to files.

Drive Labels API
Google

Use the Drive Labels API to define custom metadata taxonomies to organize, find, enforce policy, and support business processes on Google Drive files and folders.

9. Click Enable

≡   **Google** Cloud   MASbackup ▾

←   Product details

Google Drive API

Google Enterprise API

Create and manage resources in Google Drive.

ENABLE   TRY THIS API ⧉

Click to enable this API

10. After enable the API, go to OAuth consent screen at side panel.

API APIs and services

- ⁂ Enabled APIs and services
- ⬚ Library
- ⚷ Credentials
- ⠿ OAuth consent screen
- ☰ Page usage agreements
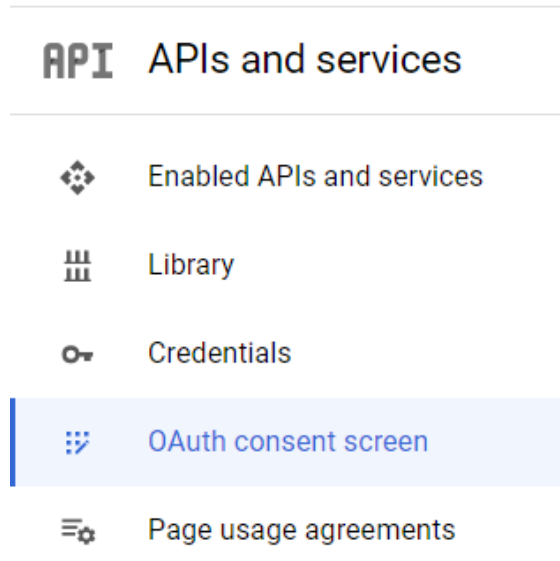
11. Enter App name and User support email.

① **OAuth consent screen** — ② Scopes — ③ Test users — ④ Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *
MASbackup

The name of the app asking for consent

User support email *
syafiqmuhammad588@gmail.com ▾

For users to contact you with questions about their consent. Learn more ☐

12. Also this Email address. Then click SAVE AND CONTINUE.

Developer contact information

Email addresses *
syafiqmuhammad588@gmail.com ✕

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE   CANCEL

13. For Scopes session, ignore and scroll then click SAVE AND CONTINUE.

SAVE AND CONTINUE   CANCEL

14. Also same with Test Users session. Ignore and click SAVE AND CONTINUE.

✓ OAuth consent screen — ✓ Scopes — ③ **Test users** — ④ Summary

Test users

While publishing status is set to 'Testing,' only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more ⧉

+ ADD USERS

≡ Filter   Enter property name or value   ❓

User information
No rows to display

SAVE AND CONTINUE   CANCEL

15. Click BACK TO DASHBOARD

BACK TO DASHBOARD

16. Then go to Credentials from side panel.

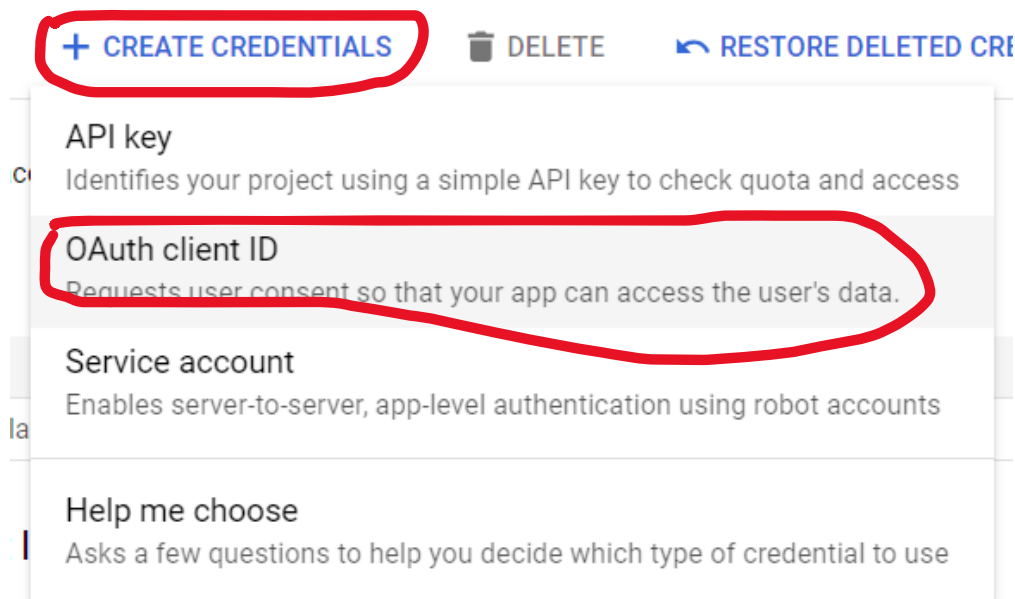**API** APIs and services

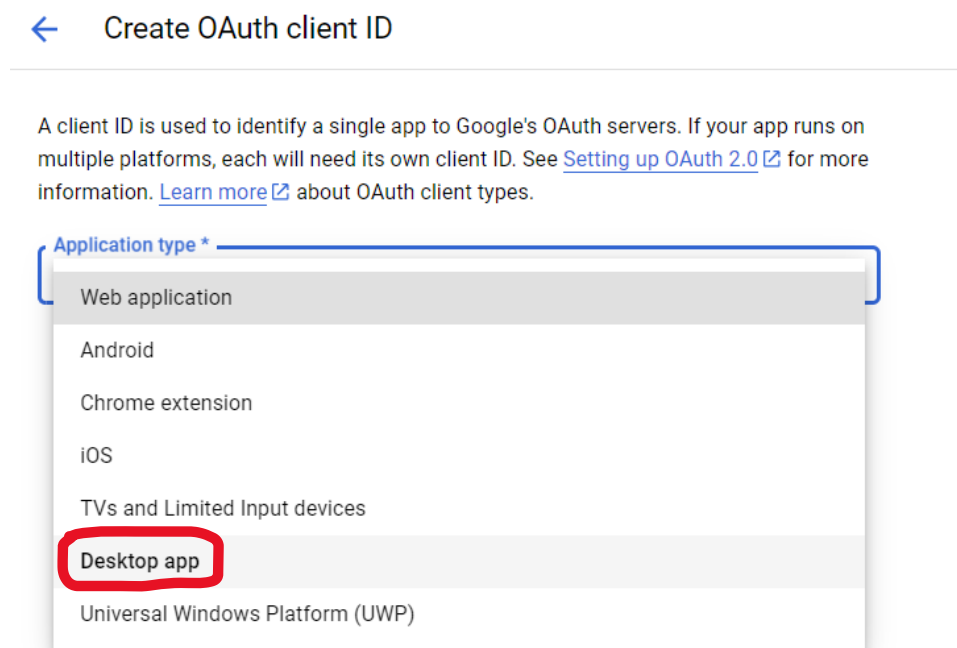⋕ Enabled APIs and services

⬚ Library

⊶ Credentials

⋮⋮ OAuth consent screen

≡☼ Page usage agreements

17. Click CREATE CREDENTIALS, then select OAuth client ID.

+ CREATE CREDENTIALS   🗑 DELETE   ↰ RESTORE DELETED CRE

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so that your app can access the user's data.

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

18. Select Desktop app for Application type.

← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 ☑ for more information. Learn more ☑ about OAuth client types.

Application type *

Web application
Android
Chrome extension
iOS
TVs and Limited Input devices
Desktop app
Universal Windows Platform (UWP)

19. Name it and click CREATE

← **Create OAuth client ID**

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 ☑ for more information. Learn more ☑ about OAuth client types.

Application type *
Desktop app ▼

Name *
MASbackup

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Note: It may take five minutes to a few hours for settings to take effect

CREATE   CANCEL

20. After created, click DOWNLOAD JSON

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

ℹ️ OAuth access is restricted to the test users ☑ listed on your
OAuth consent screen

Client ID

Client secret

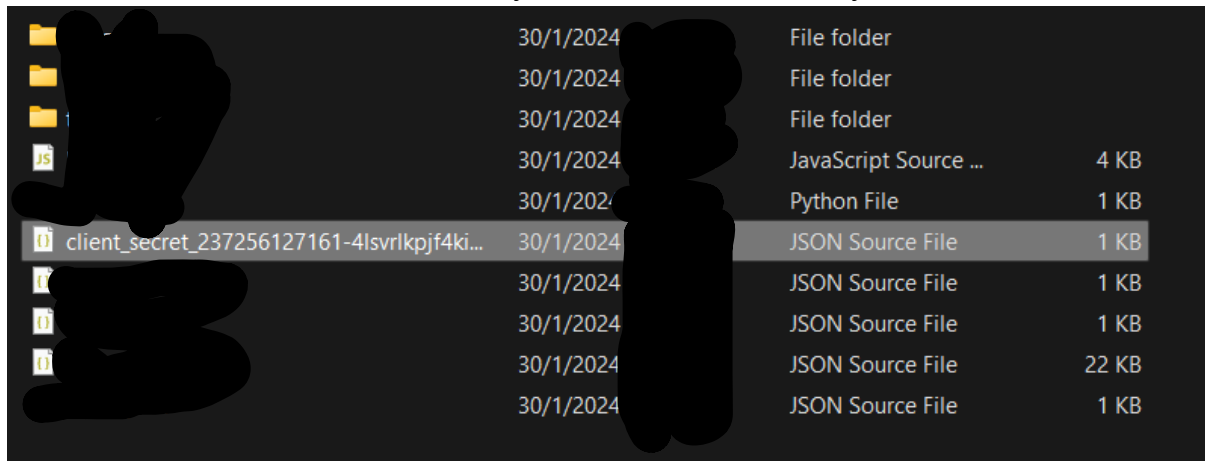Creation date   30 January 2024 at

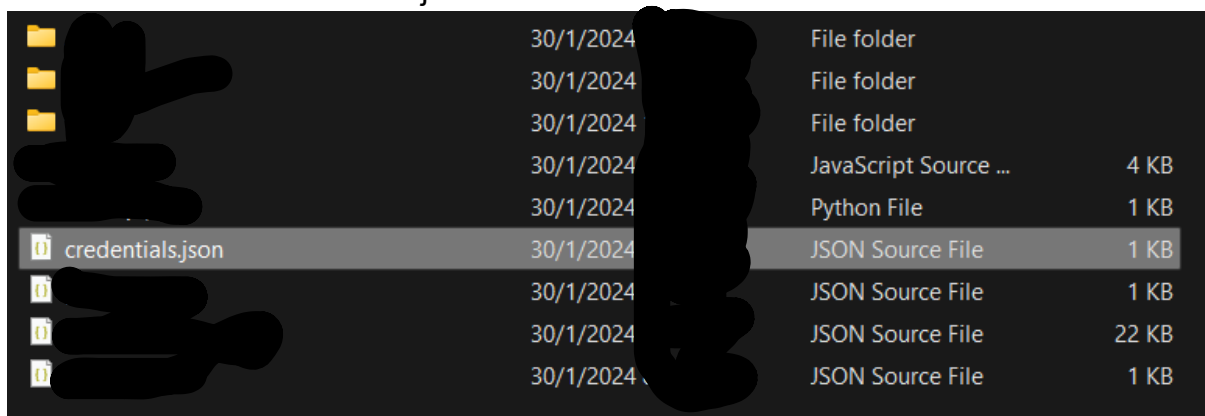Status   ✅ Enabled

⬇ DOWNLOAD JSON

21. Paste downloaded JSON file at your folder of the folder you clone from GitHub.

	30/1/2024		File folder	
	30/1/2024		File folder	
	30/1/2024		File folder	
	30/1/2024		JavaScript Source ...	4 KB
	30/1/2024		Python File	1 KB
client_secret_237256127161-4lsvrlkpjf4ki...	30/1/2024		JSON Source File	1 KB
	30/1/2024		JSON Source File	1 KB
	30/1/2024		JSON Source File	1 KB
	30/1/2024		JSON Source File	22 KB
	30/1/2024		JSON Source File	1 KB

22. Rename it as credentials.json

	30/1/2024		File folder	
	30/1/2024		File folder	
	30/1/2024		File folder	
	30/1/2024		JavaScript Source ...	4 KB
	30/1/2024		Python File	1 KB
credentials.json	30/1/2024		JSON Source File	1 KB
	30/1/2024		JSON Source File	1 KB
	30/1/2024		JSON Source File	22 KB
	30/1/2024		JSON Source File	1 KB

23. Open backup.js to modify it using any of you favourite code editor. Mine using VS Code. If you're not familiar, you can just Notepad, but it is totally not recommended. Now, go to line 100, search for "const credentials = JSON.parse(await readFileAsync('C:/Users/syafi/OneDrive/Documents/Day23/credentials.json'));"
You need to modify the path of your credentials.json file. Easiest way is to "copy as path" from File Manager, and change all \ to /
Example of mine :

```
98   async function main() {
99       try {
100          const credentials = JSON.parse(await readFileAsync('C:/Users/syafi/OneDrive/Documents/Day23/credentials.json'));
101          const auth = await authorize(credentials);
102          await uploadFolder(auth);
103      } catch (err) {
104          console.error('Error:', err);
105      }
106  }
```

24. Next, we need to modify this, at line 11, use same path as previous, BUT just replace "credentials.js" with "token.js"

```
8    const readFileAsync = promisify(fs.readFile);
9
10   const SCOPES = ['https://www.googleapis.com/auth/drive'];
11   const TOKEN_PATH = 'C:/Users/syafi/OneDrive/Documents/Day23/token.js';
12
```

25. Then, go to line 51, modify the folderPath where your persistent folder at.

```
48    async function uploadFolder(auth) {
49        const drive = google.drive({ version: 'v3', auth });
50        const folderId = '1osssI90rA08SAZeiCX7zGVso4ahSWC6r'; // ID of the folder in
51        const folderPath =                            /RenPy/Monika After Story';
```

26. For line 50, modify the folderId, get the Id from your Google Drive folder path. (You need to create the backup folder first in your Google Drive). Don't forget to save the code.
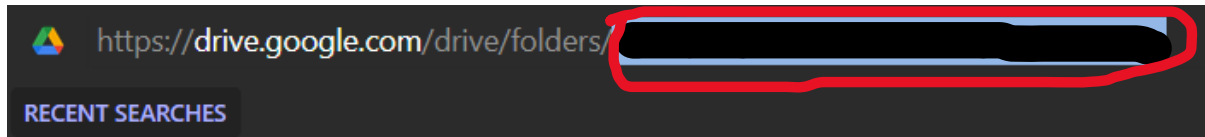
27. Execute " node backup.js ", then copy the URL to your browser for authorization request.

```
syafiq@SYAFIQLIMROG  ~\OneDrive\Documents\Day23                              [22:48]
) node backup.jsasync function main() {
Authorize this app by visiting this URL: https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=https%3A
googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost
Enter the code from that page here:ler(auth);
```

28. Choose your account.
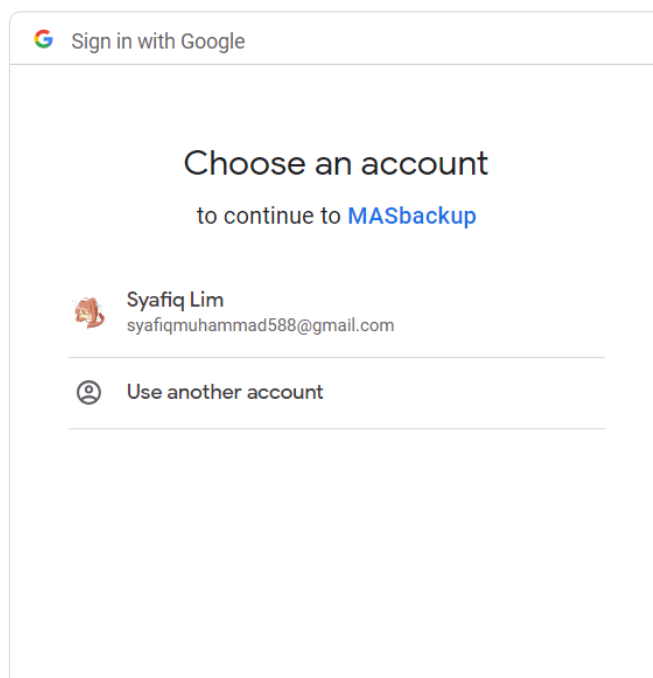
29. Click Advanced

⚠️

Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (syafiqmuhammad588@gmail.com) verifies this app with Google, you shouldn't use it.

If you're the developer, submit a verification request to remove this screen. Learn more

Advanced

BACK TO SAFETY

30. Click Go to…..

⚠️

Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (syafiqmuhammad588@gmail.com) verifies this app with Google, you shouldn't use it.

If you're the developer, submit a verification request to remove this screen. Learn more

Hide Advanced

BACK TO SAFETY

Continue only if you understand the risks and trust the developer (syafiqmuhammad588@gmail.com).

Go to MASbackup (unsafe)

30. Click Continue

G Sign in with Google

MASbackup wants access to your
Google Account

syafiqmuhammad588@gmail.com

When you allow this access, MASbackup will be able to

See, edit, create and delete all of your Google Drive files. Learn more

Make sure that you trust MASbackup

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

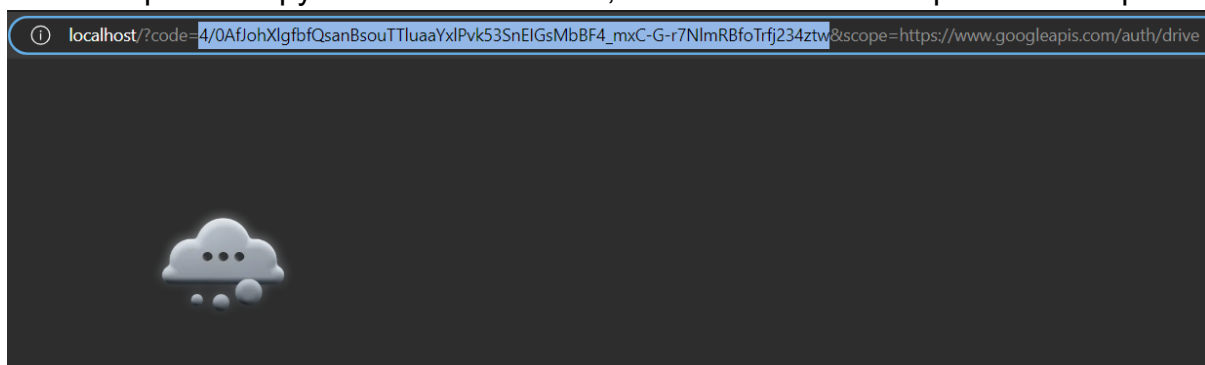Learn how Google helps you share data safely.

See MASbackup's privacy policy and Terms of Service.

Cancel   Continue

31. Don't panic. Copy the code at the URL, after code= until &scope. For example :

localhost/?code=4/0AfJohXlgfbfQsanBsouTTluaaYxlPvk53SnEIGsMbBF4_mxC-G-r7NlmRBfoTrfj234ztw&scope=https://www.googleapis.com/auth/drive

32. Go back to terminal/cmd/powershell, and paste the code.

```
> node backup.js
Authorize this app by visiting this URL: https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=https%3A
%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&response_type=code&client_id=237256127161-4lsvrlkpjf4kimkiu9gg0qj4rhovsi4t.apps.
googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost
Enter the code from that page here: 4/0AfJohXlgfbfQsanBsouTTluaaYxlPvk53SnEIGsMbBF4_mxC-G-r7NlmRBfoTrfj234ztw
```

33. That's it. It will upload to your Google Drive folder. After this you don't need to this authorization again, just execute "node backup.js" and it will replace files and backup.

```
Uploading [═══════════════════════════] 100%

Folder uploaded successfully.
```

e. Now, for submod (Ren'Py) part

1. Copy the mas_backupGD.rpy to the your submods folder in your DDLC (I'm pretty sure you know what I mean if you have installed any submods before).

2. Open mas_backupGD.rpy using any code editor to modify it. Go to line 30, modify the js_file_path to the backup.js file of the file you previously did. Save the file.

```
27    label testing_syafiqlim:
28        python:
29            # Specify the path to your JavaScript file here
30            js_file_path = "C:/Users/syafi/OneDrive/Documents/Day23/backup.js"
31
32            # Call the function to execute the Node.js script
33            node_result = execute_node_script(js_file_path)
34
```

3. That's it, now you can get the option in Be right back… menu.
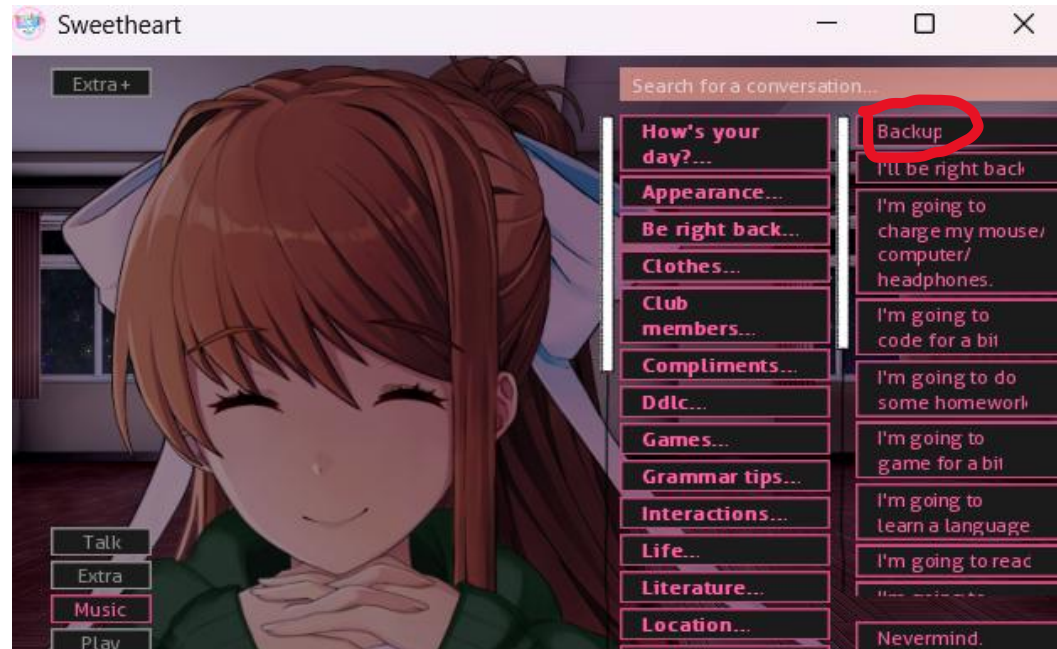
WARNING : You don't need to delete folder if you want to renew the backup, just run the backup tool and it will automatically replace the old one with new one. Otherwise, you need to create new folder and paste new folderId in backup.js code.