amiibo Automate Music Player

If you're an avid amiibo collector, you may have come across "amiibo Music Stands" like this one: https://www.youtube.com/watch?v=iGOuCgbN1nc, built by people to get more mileage out of their figures. You may have even dreamed about having one for yourself, but held back due to your lack of knowledge of soldering or coding.

Well, seeing as most phones have NFC scanning capabilities, I wondered if it was possible to make a similar version using my own phone, and it turns out, the answer is yes! Even better, it can be done in such a way that the code is easily available to anyone running android, and with just a bit of file editing, soon even you can be jamming out to your favourite tunes using your beloved figures.

How it works

If you didn't know, an amiibo's data is held in a NTAG215 chip within the amiibo's base. Each tag possesses a serial ID, unique to that tag alone. Although an amiibo's data is proprietary, and not easily decoded without additional software, the serial ID is still easily readable by any NFC reader, including a NFC compatible phone.

The main brains of the music player is an app called Automate by LlamaLab (https://play.google.com/store/apps/details?id=com.llamalab.automate&referrer=utm_source%3Dhomepage). It is an app that allows one to automate the functions of one's phone, allowing one to create "flows" to do a task comprised of "blocks" that tell the phone to do something, including the scanning of a NFC chip. Additionally, Automate also allows the sharing of flows to the community, allowing a creator to share their work with the rest of the world.

For our purposes, I have created two flows in order to convert our android phone into an amiibo music player, being "amiibo ID List", which allows you to scan your amiibo and have Automate record all their IDs to a .json file alongside a sample pathway for the audio, and "amiibo Music Player", which will read the .json files to play the audio files.

The Technical Breakdown

In truth, all the amiibo Music Player flow does is read the amiibo's serial ID, and then tell the phone to play three audio files one after the other; being the announcer voice clip, the character's voice clip, and then the music. Since each amiibo's ID is unique, the phone will always play the correct files (assuming the ID is set to the correct pathway in the .json files).

Unfortunately, the unique serial also means that every person will have to create their own list of IDs to play music to, but I have attempted to make the process for doing so as streamlined as possible.

If you choose to use a non-smash amiibo, or simply want to disable the announcer or voice clip, then you can use the "null" audio file in place of any other. Null is simply a short (0.5^{th} of a second) clip of silence, and after it finishes, Automate will play the next audio file as intended, essentially 'skipping' that clip.

Preparations:

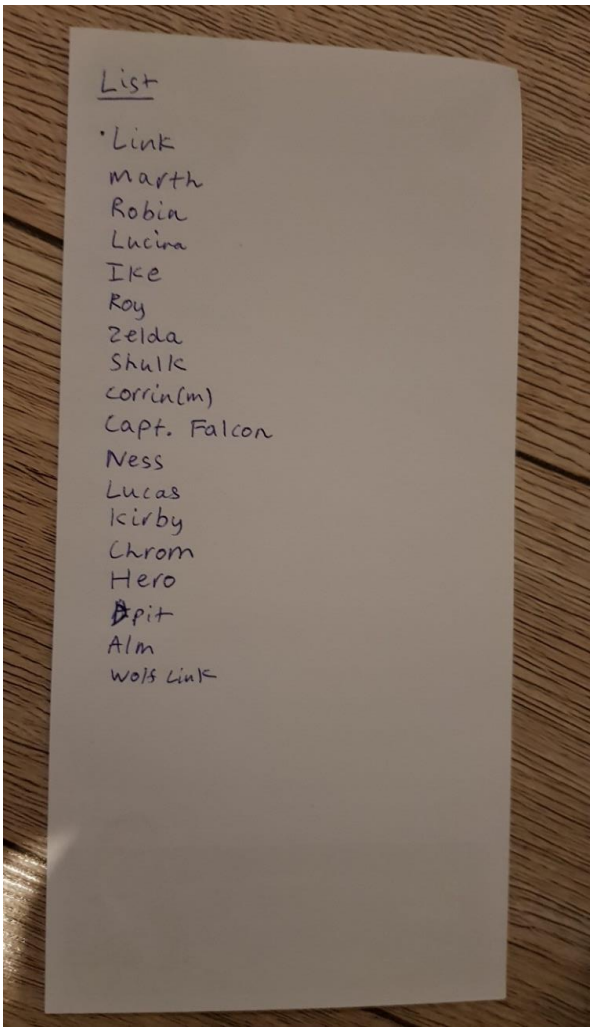
You will need:

- An NFC capable phone
- Automate app

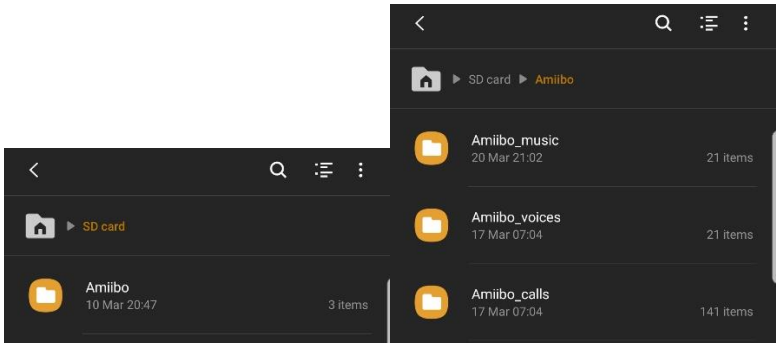
- App that plays music files (I use Musicolet)
- Code editing software (notepad or notepad++ work fine)
- Pen and Paper
- A computer
- A USB Cable to connect your phone to computer

Steps:

1. Before doing anything, set your phone's timeout to a long time (10 minutes or more), otherwise your phone will lock itself and stop scanning NFC tags, which can easily happen if you listen to multiple amiibo songs at once. You can also download an app to keep your phone's screen on indefinitely.
2. Create a list of your amiibo to scan, and write the order in which you will scan them. This is essential! You need to do this so you don't accidentally assign the wrong ID to the wrong audio files later!

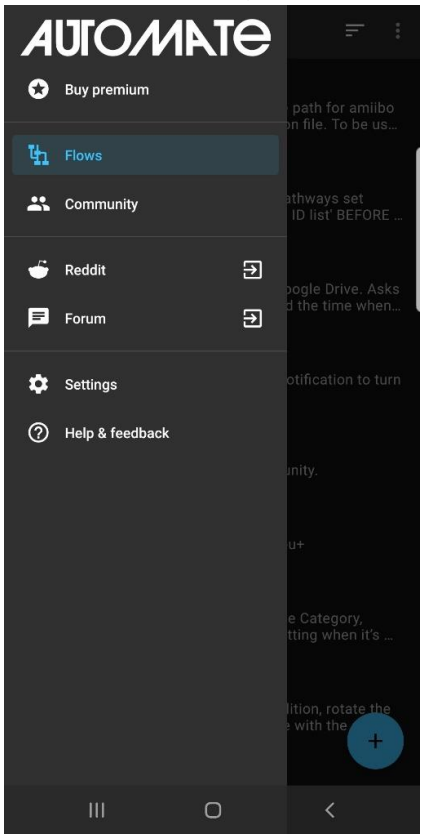


3. Download the "amiibo" folder from my github (https://github.com/Jinraider/amiibo-music-player.git) and put it somewhere convenient your phone (I chose the SD card root).

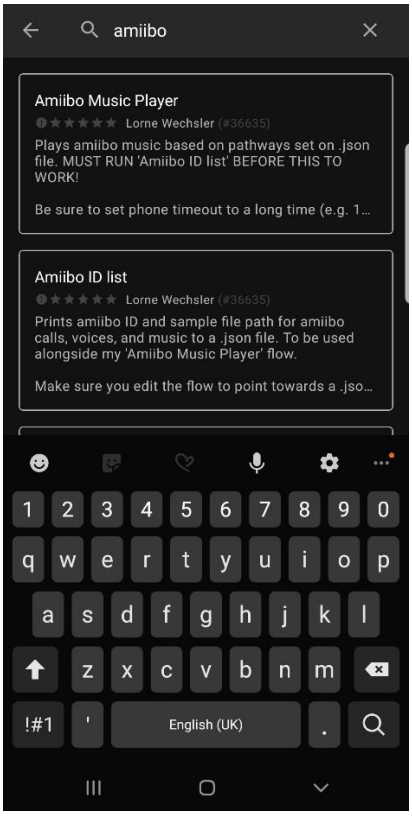


- Before continuing, take a moment to look through the “Amiibo_calls” folder, and make sure you understand which files corresponds to your desired characters; you will need to match these names exactly to your amiibo IDs for the music player to work!
- If the voices and music folder don’t have files corresponding to your desired amiibo character (or you just dislike my choices), you will need to add your own to for the phone to play.
  - Add a voice clip for your character to amiibo_voices, and name it [character_name].wav (if your character is silent, skip this).
  - Add a music clip to amiibo_music, and name it [character_name].mp3 (you *can* use a different format, but you will need to remember the extension for later.

4. Download Automate and access community on the sidebar (it will ask you to download an add-on to allow it to access the internet, so do so).

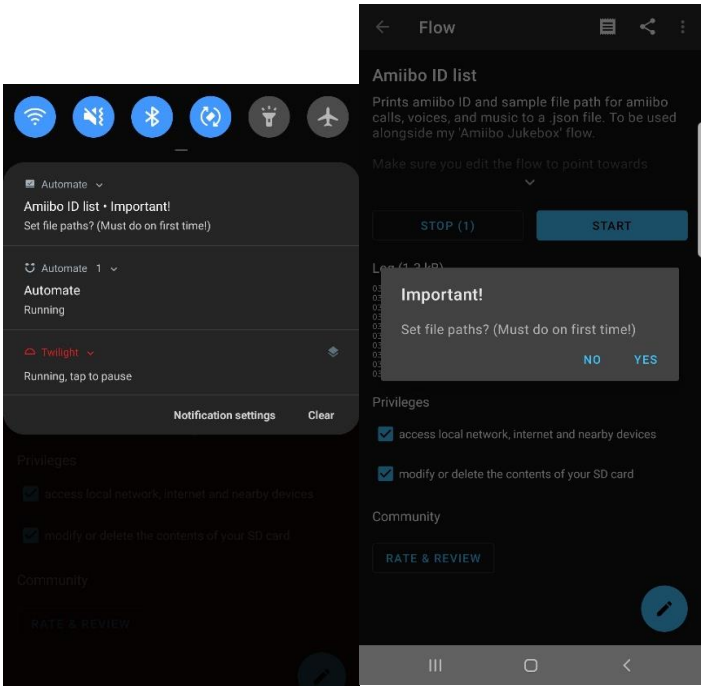


5. On community, search for “Amiibo ID list”, and “Amiibo music player”. Download both flows.



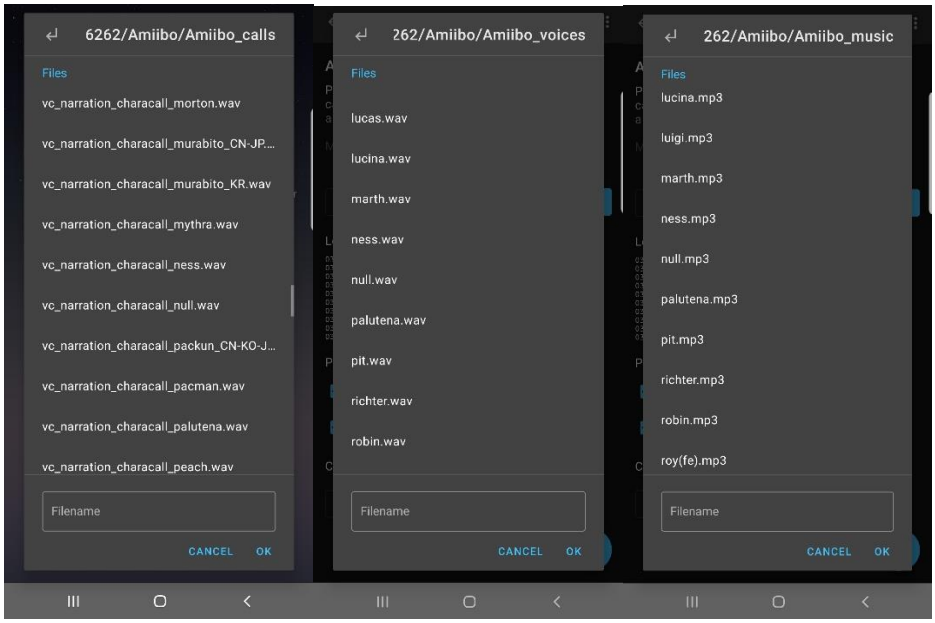
Not my actual name, FYI

6. Enable your phone’s NFC and run the “Amiibo ID list” flow. After pressing ‘start’, pull down the phone’s dropdown to open the Automate pop up. It will ask you to set a sample file destination; select “yes”.



You will also need to reset the file paths if you ever edit the flow

7. Automate will then ask you to pick sample file pathways for the announcer, voice, and music clip. It will take the pathway to these samples and assign the amiibo ID to them for us to change later. For now, find the “amiibo” folder you put on your phone earlier, enter the “amiibo_calls” folder, and find the one that is called “null”. Do the same for the next two pop ups, for the “amiibo_voices” and “amiibo_music” folders.



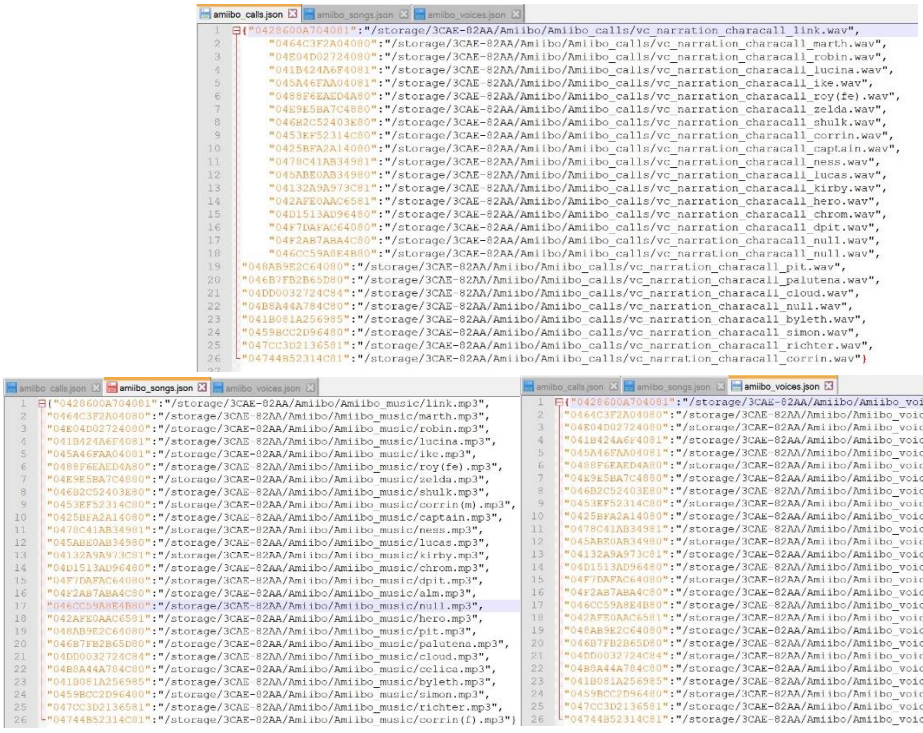
8. After selecting the files, you are ready to scan your amiibo! Scan them based on the order you wrote earlier. Your phone should make a chime or vibrate to confirm that an amiibo has been scanned. Once it does, you are free to scan another amiibo. Once all your amiibo have been scanned, stop the flow by pressing the ‘stop’ button.

9. Connect your phone to your computer, and access the root of your phone’s internal storage. Copy “amiibo_calls.json”, “amiibo_songs.json” and “amiibo_voices.json” to your pc. These are the files that Automate created; they contain your amiibo’s IDs and the sample audio pathways. Open up the three files in a code editor like notepad or notepad++.

10. **Important: Pay close attention to this step and your file pathways! Assigning incorrect pathways is the most common way of getting an error with your music player!**
Before doing anything, put a left curly brace, {, before the first amiibo ID on your first line of code. Next, **delete** the final comma after the final pathway of your final line of code, and replace it with a right curly brace, }. Do this for all .json files. This will make the .json file readable by Automate.

    Replace the ‘null’ in each .json with the name that matches the audio files you want to play for that character. You can use CTRL+H to “Find and Replace” and make this simpler. The name must match exactly, extension included. If you wish for the player to play a file in a different format, you can delete the .mp3 and replace it with the file extension of your choice. You will need to do this for all .json files.

    After finishing, your code should resemble something like this:



Remember that "null" is simply a silent music clip.

11. Save your .json files. Copy them back into the internal storage of your phone (keep a copy on your computer in case your code isn't right); when asked, choose to 'replace' all the ones you had earlier.

12. You should now be ready to go! Start up the "Amiibo Music Player" flow, scan an amiibo, and jam out!

When you want to scan another amiibo, simply tap the amiibo again to stop the music. Scan your next amiibo and repeat!

Press stop when you wish to stop the music player altogether.

13. If you wish to add more amiibo to play music with, simply start the amiibo ID List flow again (you won't need to set a file pathway this time), and start scanning your amiibo.

Recopy "amiibo_calls.json", "amiibo_songs.json" and "amiibo_voices.json" back onto your computer (if your music player worked, you can safely overwrite the versions on your computer), replace the right curly brace, }, with a comma, and put the right curly brace after the final line.

Again, just make sure your code resembles the example above, and you should be good.

Troubleshooting

If you unfortunately got some errors (Automate will notify you when it does. Be sure to read it to see what it says), you may need to recheck your .json files.

Common errors are:

Incorrect file pathways

- Automate can only open audio files if they match the file pathway in the .json file exactly, extension included. Make sure the name of the file and what is in its respective .json match, and be careful if you choose to use audio files in different formats (especially for songs).

Invalid formatting in .json file

- Pay special attention to your .json files. Make sure every line (except the last one) is ended with a comma, and the last line is ended with a right curly brace }. Again, compare with the examples above, and make sure that yours look similar.

So What's Next?

As previously stated, you can keep adding new amiibo IDs to your list to tell Automate to play a certain music file for any one of your amiibo, so long as you have the music file on your phone.

But if you looked on GitHub, and noticed it didn't have a music file for your favourite amiibo character, why not contribute to the repository? Simply create an account, add a voice clip and music file to the respective folder under the "new music" branch, and I will add it to the main file list. Every little contribution, even for a single character, helps!

Special Thanks:

The makers of Automate, for giving me the tools to put this together.

ALL the guys over at r/AutomateUser for helping me understand Automate and its functions, but especially u/ballzak69, u/B26354FR and u/ElC1d; I quite literally couldn't do this without them.