Guide to Making a Meower Scratch Client

TABLE OF CONTENTS
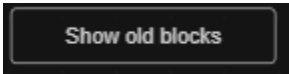
DEPRECATED BLOCKS

!! As of **November, 2023**, some **CloudLink 4** blocks have been deprecated and hidden. Blocks that were hidden were also used for **Scratch clients**. But don't worry! You can still make a client and with those blocks too. Just scroll all the way down to the bottom of the extension's block panel, and you'll see a tab called "show old blocks" and if you click it, you'll get the blocks back. !!

Show old blocks

BOT TEMPLATE

The bot template used for this guide was created and owned by @zz on Meower. It is used with credit, and with permission from said owner. You are free to use this guide **with given credit**. Under no circumstances, may you use this guide without proven and stated credibility given towards us.

If you would like to get in contact with @zz on Meower, you can find them here:
https://app.meower.org/users/zz

GUIDE

This guide was entirely created by @LordPenguin a.k.a @lpxyza with credit to @zz for the bot template.

This guide is protected by the Apache License 2.0 and you may not use any part of this guide without given credit.

If you would like to get in contact with @LordPenguin on Meower, you may do so by any of the links below:
https://app.meower.org/users/LordPenguin
https://www.roblox.com/users/275827689/profile

PREPARING THE PROJECT

---

¹ | i.e. TurboWarp or PenguinMod

² | if the extensions gallery **does** contain said extension

³ || if the extensions gallery **does not** contain said extension

To prepare the project to become a *Meower Client* you must first install the **CloudLink 4** extension on a ***supported Scratch mod¹*** of your choice.

1. Open [*TurboWarp*](#) or [*PenguinMod*](#)
    For PenguinMod-
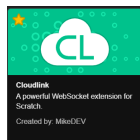
    > When opened, press "Create" on the top left.
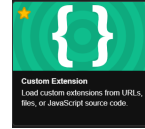
2. Select the *Extensions* button on the bottom left.

3. Import the **CloudLink 4** extension.
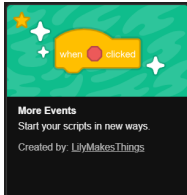   **If CL4 is an option in Extensions²-**

    If you need to import CL4 manually³-

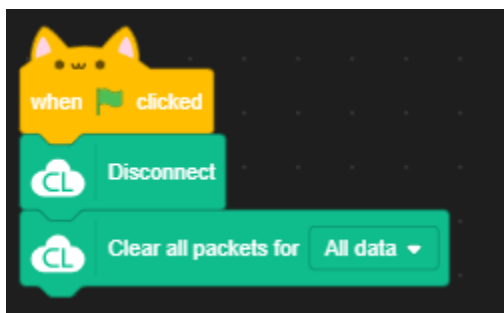Use the "Custom Extension" and import the .js as a file.

4. **(Optional)** Import the **More Events** extension.

RESETTING THE VARIABLES

At the start of every project, if it's technological and contains changing variable(s), it is recommended to **reset** the variable(s) before usage.

Make sure to include this script at the start of the project-

This script will-

    (1)   disconnect the client from the server

    (2)   clear all received/sent packets

If you want a cleaner way to reset everything, it is recommended to create a **custom block** labeled "Reset" and put all of the reset blocks underneath it. Then on flag click, the first thing you want it to do is to trigger the "Reset" block.

CONNECTING

Now is on to the more technological part: connecting to the server.

1. You'll need to make **three** custom blocks. One labeled "Connect" another labeled "Connect (retry)", and the last one labeled "start bot".

2. For both of the "Connect" blocks, attach this script underneath the {define{s.
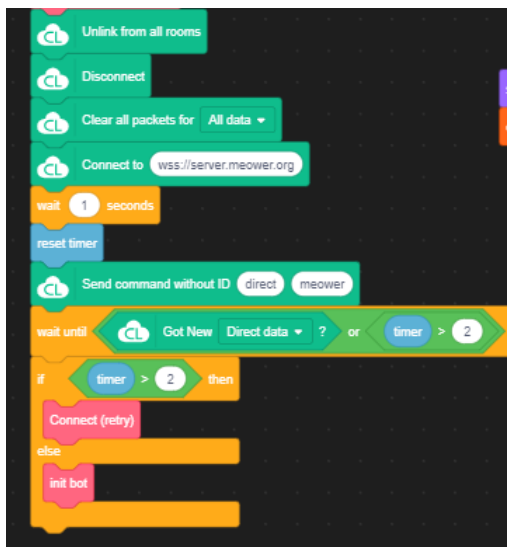
These scripts are the main connection scripts. You'll want to add the **Connect** block right under the **Reset** block.

The Meower server address is "wss://server.meower.org".

INITIATING THE BOT

If you've followed the instructions correctly so far, the next step you need to do is start the bot. You do this by sending a command to the Meower server and logging in to an account.

1. Attach this script underneath the "start bot" definition.
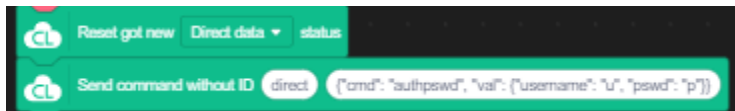
This script will login and authorize your credentials for connection. In the second input, look for the JSON "username" and replace "u" with your Meower username. Do this again but with the JSON "pswd" and replace "p" with your Meower password.

After doing that, you need to attach this script right underneath the previous script. The "logged in?" variable is optional.

*When the timer is > 2 it reads as a timeout error and doesn't connect.
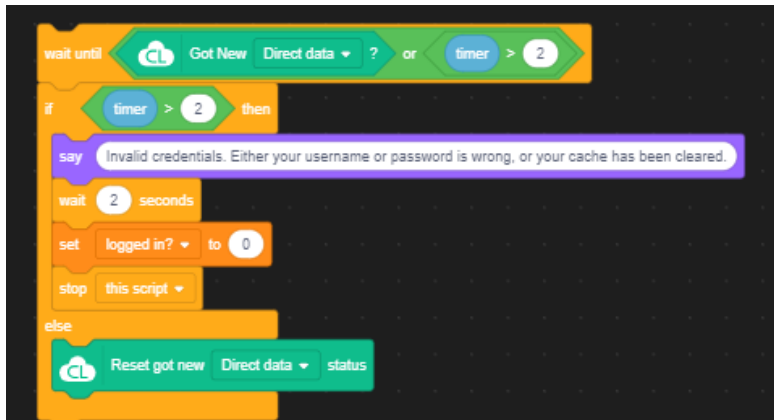
Hooray! You've completed the **INITIAL SETUP**!

SENDING MESSAGES

Okay, you have now **connected, started,** and **logged in** to Meower from **TurboWarp**. The next step in this guide is "sending messages" which is pretty essential if you actually want to *use* your client.
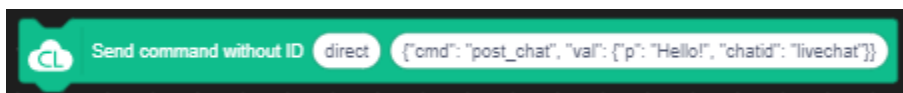
1. To actually send a message to the Meower server, you need to use this script:

   Replace the "Hello!" with anything you want to say and it'll send it to the home page.
   However…

2. If you want to send a message to a group chat that the currently logged in account is in, or send a message to livechat you need to use this script:

   Livechat's ID is "livechat" and if you want to connect to a group chat, you have to put the "chat-id" (found in the Meower group chat, on the top) into the slot where "livechat" is. *You should include a forever{ask (), send (answer)}.*

DELETING MESSAGES

If you want to delete a message that you (using the client) made, you can use this script:

```
CL  Send command without ID  direct  {"cmd": "delete_post", "val": "ID"}
```
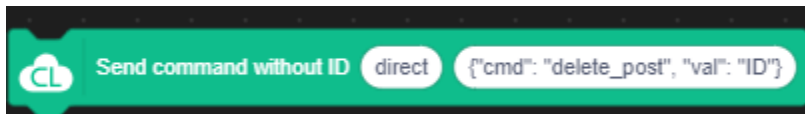
Replace the "ID" with the post-id found in "direct data" BUT you can only delete the messages **you** send. Which is why it's recommended to have some way to store all of your posts' ids. Preferably in a list, where you can then easily find them.

RECEIVING MESSAGES

Now onto one of the more essential parts of a client. Actually receiving the messages people send (including you). Use the script below to store new messages in a "chat" list:

```
CL  When I receive new  Direct data ▾  message
if  not  CL  state  of  CL  Direct data  =  101  then
    set  new post ▾  to  join  CL  u  of  CL  Direct data  join  :  CL  p  of  CL  Direct data
    insert  new post  at  1  of  chat ▾
```

But something that is a downside to the whole receiving the messages thing is that some messages are sent from a bridged device. Their bridged device name is stored in their post username, and instead, the username that sent it is stored in the actual post. So in the case of a bridged message, you should modify the previous script to this:
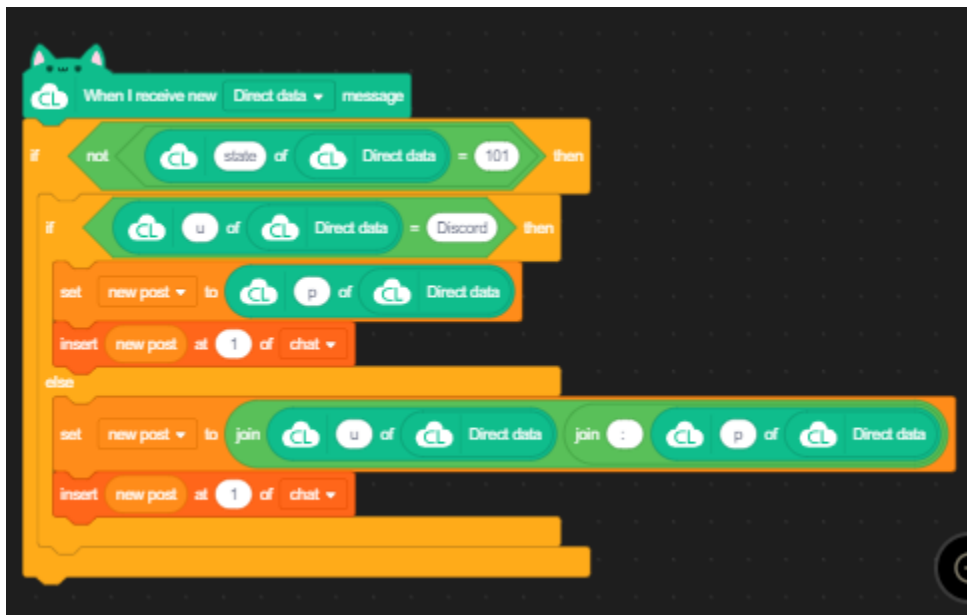

If you followed the steps correctly, you should now be able to see bridged messages without that annoying tag in the way.

FETCHING USER DATA

If you want to fetch a user's statistics like the "whois" command frequently used on most bots, you can do so by this block:

Just change the URL to "https://api.meower.org/users/" and at the end, put the username of the person you want to fetch the stats of. Like this:
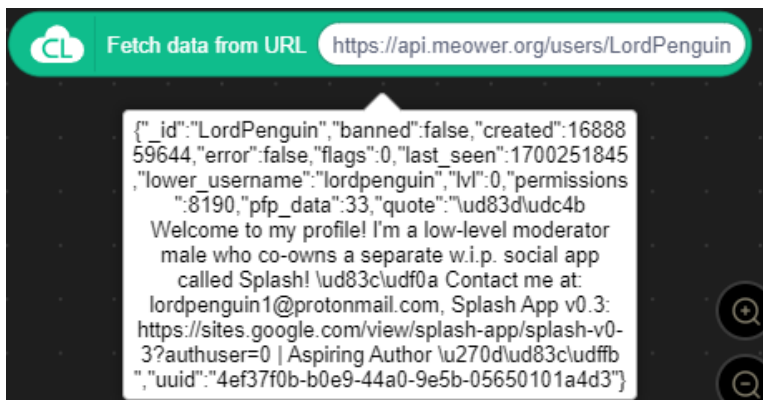
FETCHING MEOWER STATS

It's simple, really. Instead of the "/users/" part, replace it with "/statistics" like "https://api.meower.org/statistics".

FETCHING MEOWER STATUS

This time, you'll change the end parameter to "/status" for fetching the Meower server status. This helps to check for downage, outages, etc.

SEARCH MEOWER POSTS

This is where it gets a little more complicated. You'll need to change the parameter to "/search/(room)?q=(query)" like this:
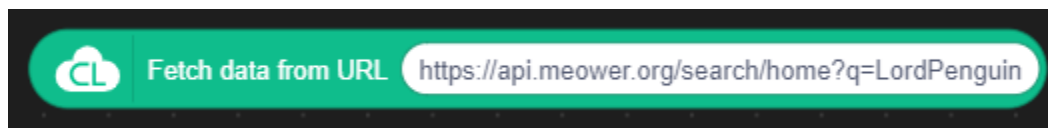
Just replace (room) with the chat-id you want to search in, and the (query) with the search query.

CONCLUSION

This is all the basic information you need to know. Now the rest is up to you! What are you going to add? A bot with command detection? A link to group chat(s) function?

You can DM @LordPenguin or @zz by the following links:
https://app.meower.org/users/LordPenguin
https://app.meower.org/users/zz

END OF GUIDE