

Your favorite catgirl's projects

- [26-03-30: Cables.GL, what's that ?!](#)
- [26-03-06: Kwutty, my public instance of kuty to shorten links](#)
- [26-03-04: listening to banger song as i'm writing this](#)
- [26-02-23: Waaaa, s t u f f](#)
- [25-07-16: me like music and cool visuals](#)

26-03-30: Cables.GL, what's that ?!

Good morning beautiful people reading this !

I often mention cables.gl, but what is it ????

Well here's what their website says :

“ Visual Programming with cables

Cables is a tool for creating beautiful interactive content. With an easy to navigate interface and real time visuals, it allows for rapid prototyping and fast adjustments.

You are provided with a set of operators, such as mathematical functions, shapes, materials and post processing effects.
Connect these to each other with virtual cables to create the experience you have in mind.

Easily export your piece of work at any time. Embed it into your website or use it for any kind of creative installation.

I will not get too technical in this blog post, gonna be short and mainly to showcase stuff i already made :3

Where's VVVV ?

In my first blog post here i used VVVV; And i used it quite a bunch more after that! very cool software.. But after switching completely to linux well... it just not worth the effort to get it running smoothly, every solution i tried resulted in at least one really annoying bug that made the experience a pain.

Searching for an alternative

So i wondered, is there anything else like VVVV ?

I tried a few, puredata/plugdata - TouchDesigner - others you can find on here :

<https://alternativeto.net/software/vvvv/>

Then i came across cables.gl, at first redo-ing my project on there wasn't really straight forward, linkage order between nodes is much more important than in VVVV, and for some, a bit less intuitive (thanks to the online examples and ressources i did it!)

In 4V i was able to use the sound output of the pc directly, AFAIK there is no easy way of doing it in cables.gl, i could do a sink that sends it back to an input but am lazy and so i just went with the "media player" approach at that time

First project, music visualiser with a fixed music and text

https://public.frezlotte.ch/cables/music_visualiser_old/

There's a lot of keybinds u have to know to use (Q-A; W-S; E-D; clicks and scroll), and parameters are not clamped (stopping to a min or a max) so u can easily make it look like crap or stop rendering, but as a first project, not bad :3 Also it made me learn that not everything translates well on a browser, for some reason at that time the multi-ports nodes worked fine in editor but broke everything in a webpage, really painful for a newcomer.

Second project, new year clock

https://public.frezlotte.ch/cables/year_countdown/

New year was coming fast and i wanted to do something, either a mini game / activity or idk, something

For some reasons i won't go into details about, i wasn't sure with who i would spend my NY eve, but shortly after getting an idea of some "plans", i got to work on the timer to put on a tablette somewhere

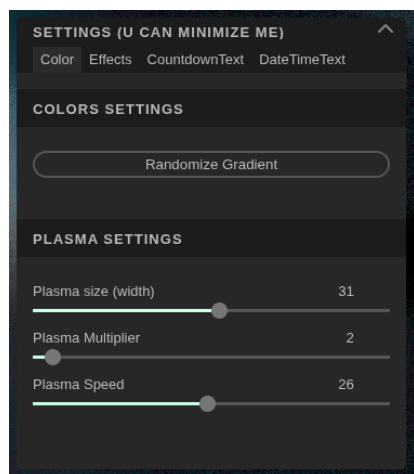
At first i really wanted to put a rick roll or SOMETHING at 00:00, that would had been fun, but i ran out of time and just made a quick "HNY" and made the background go a tad faster, that's it.

Looking back a lot of ppl used it so i would had felt bad rickrolling all their families... i won't do it this year ! no promises tho :3

Second project, the sequel

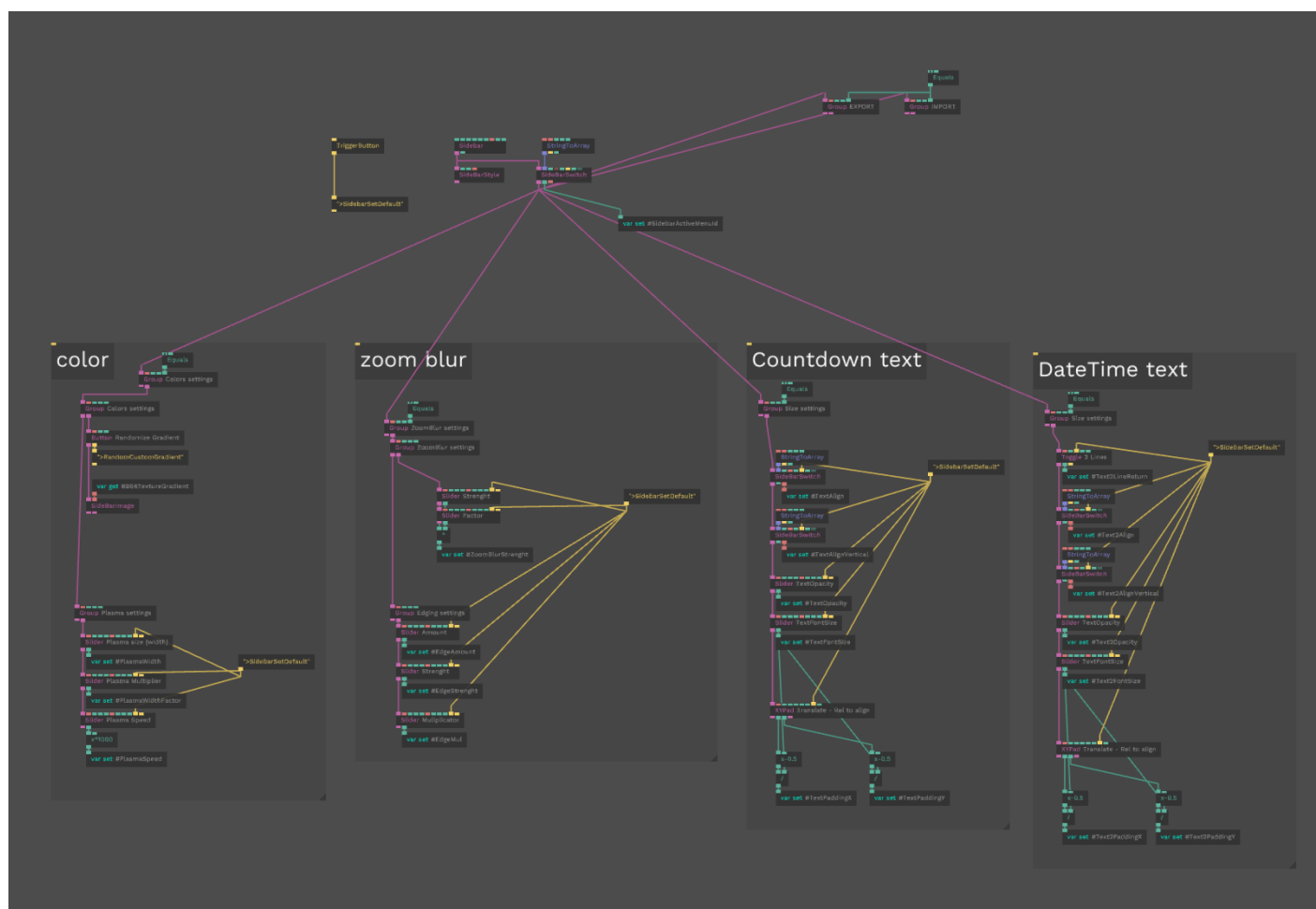
Worked well for new year, but i wasn't gonna stop there, it's a gimmick but the background was really pretty, even at that time

i wanted more control and settings, and i came across the notion of sidebar



This little fella, you can toggle it at anytime top right of the browser, in there i made multiple pages, and tbh, am quite happy with how it is right now

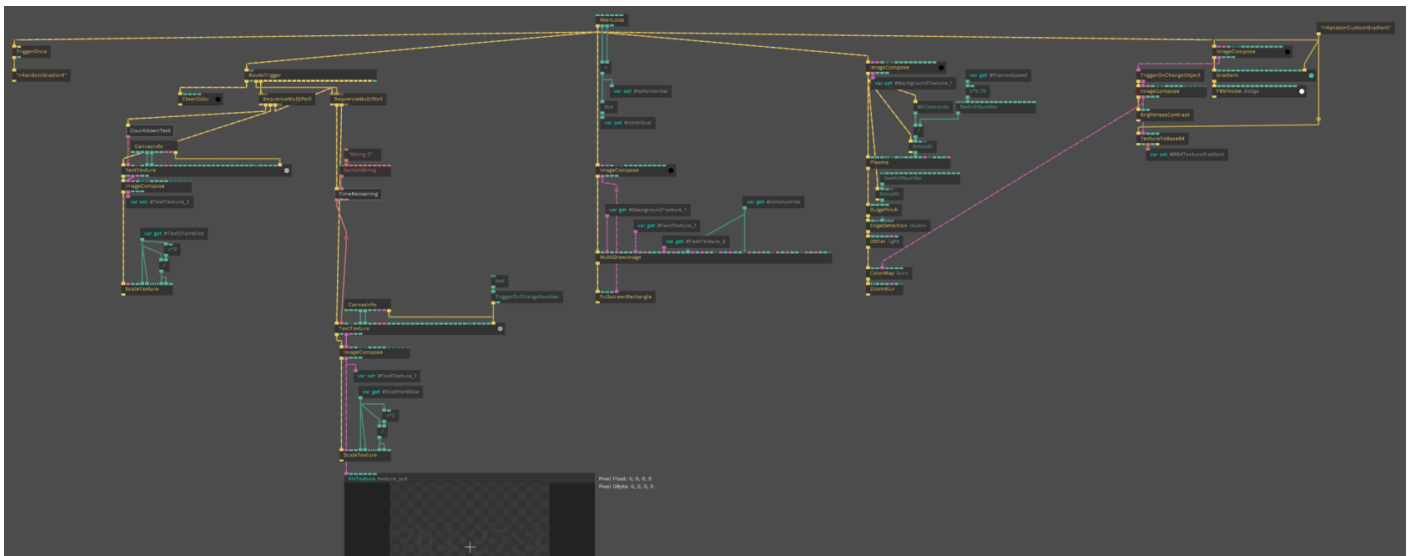
Using it i was able to find some very pretty settings really fast, using vvvv i thought of it as a menu you did after the fact, but in cables.gl, having it along the making of something is really useful to be prototyping in no time



Sorry for the mess ! Here's the menus in node form, that's the entirety of the sidebar right there.

Top right is a future plans, import and exports of settings.

Here's for the visual part of it



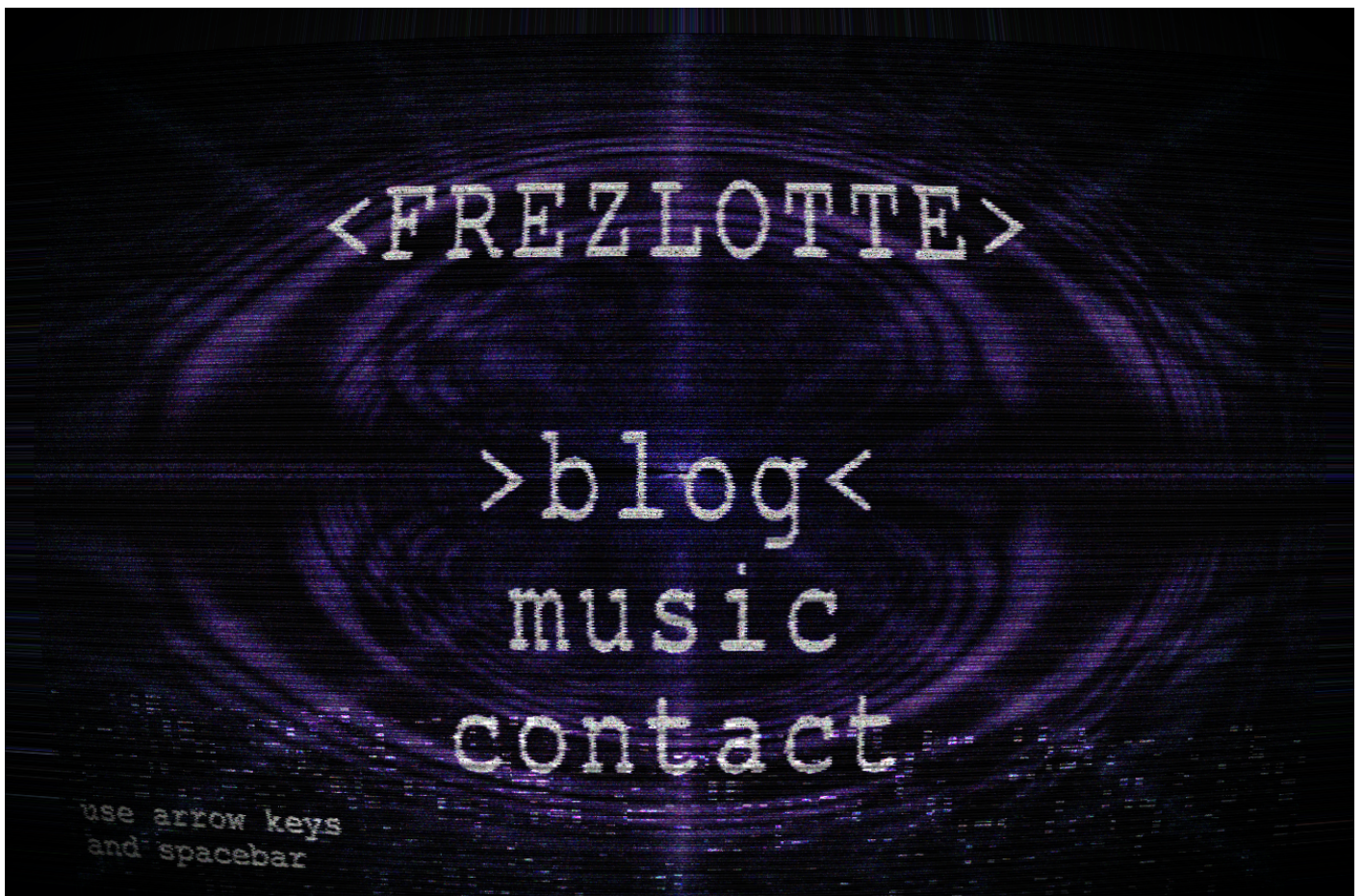
Seeing it again today, i would changes a looooot of things, but hey it means i progressed so yay

Performance wise you can bet it runs like a burning trash can. Optimising ??? Yo girl that's not a real word what are you inventing rn ??

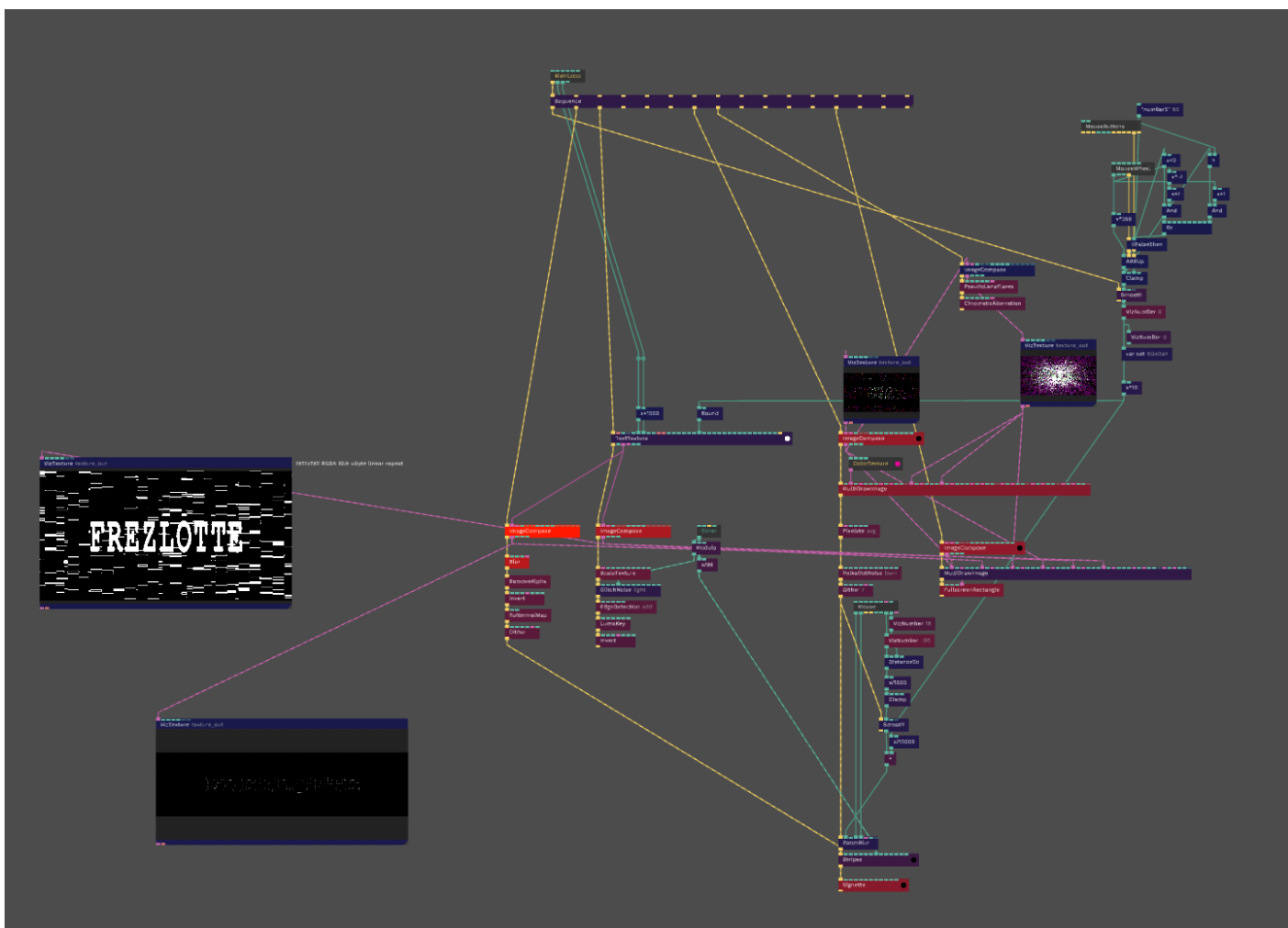
Third project, frezlotte.ch

I always wanted a fancy, artsy landing page for <https://frezlotte.ch>, and now, i got one !

My latest project in date, went through a number of iterations, but am very happy about the end result !



The glitches, the tv effects, the visual, exactly like i pictured it in my mind before actually doing it, maybe even better :3



26-03-06: Kwutty, my public instance of kutty to shorten links

kwutty - URL, free public URL shortener

Someone on discord spoke about targeting a qr-code to as linktree so that we are sure we can always make sure the target is available and we don't need to reprint some cards.

So it got me thinking, i'm currently doing annoying HTTP only redirect that ppl often open as HTTPS and gets a warning for, plus it's not very readable (and it's publicly available on site like <https://crt.sh/?q=frezlotte.ch>) so i digged a little using my trusty <https://alternativeto.net> to find a self hosted link shortener service, with telemetry, moderation, and the possibility to make link password protected.

came across <https://kutt.to/>, very well build and easy to setup, the compose.yml is as follow :

```
services:
  kutt:
    build:
      context: .
    volumes:
      - ./db_data_sqlite:/var/lib/kutt
      - ./custom:/kutt/custom
    environment:
      DB_FILENAME: "/var/lib/kutt/data.sqlite"
    networks:
      - ${T_NETWORK:-tproxy}
    labels:
      - "traefik.enable=true"
      - "traefik.docker.network=tproxy"
      -
      "traefik.http.routers.${T_SERVICE_NAME}.rule=Host(`${T_SERVICE_SUBDOMAIN}.frezlotte.ch`) ||
      Host(`${T_SERVICE_SUBDOMAIN}.zpy.ch`)"
      - "traefik.http.routers.${T_SERVICE_NAME}.entrypoints=${T_ENTRYPOINT:-websecure}"
      - "traefik.http.routers.${T_SERVICE_NAME}.tls.certresolver=myresolver"
      - "traefik.http.routers.${T_SERVICE_NAME}.service=kwutty"
```

```
- "traefik.http.routers.${T_SERVICE_NAME}.middlewares=${T_MIDDLEWARES:- bouncer, umami,
umami-feeder, anubis}"

- "traefik.http.services.${T_SERVICE_NAME}.loadbalancer.server.port=${T_SERVICE_PORT}"

networks:
  tproxy:
    external: true
```

Like the usual, i'm using [traefik](#), with all my usual middlewares

I'm probably add a ipallowlist middleware on JUST the admin path but i'm not sure how doable it is yet

There's a umami option in kutt but i decided to not use it since i have a umami-feeder (automatically register websites in umami) so the work was already working.

The admin page listing link looks like that :

Recent shortened links.

Total links: 3 Links Users Domains

Search link... Search domain... Search user...

Banned... Anonymous... Domain...

10 20 50 | < >

Original URL	Created at	Short link	Views
https://discord.gg/Qm6yXq2bEv by (view links)	15 hours ago	/discord s.frezlotte.ch	0
https://n8n.frezlotte.ch/form/7766c9d4-aff1-452a-b455-a50 by (view links)	15 hours ago	/music s.frezlotte.ch	2
https://n8n.frezlotte.ch/form/21a7eb1a-1278-4c57-96f5-b47 by (view links)	15 hours ago	/contact s.frezlotte.ch	1

10 20 50 | < >

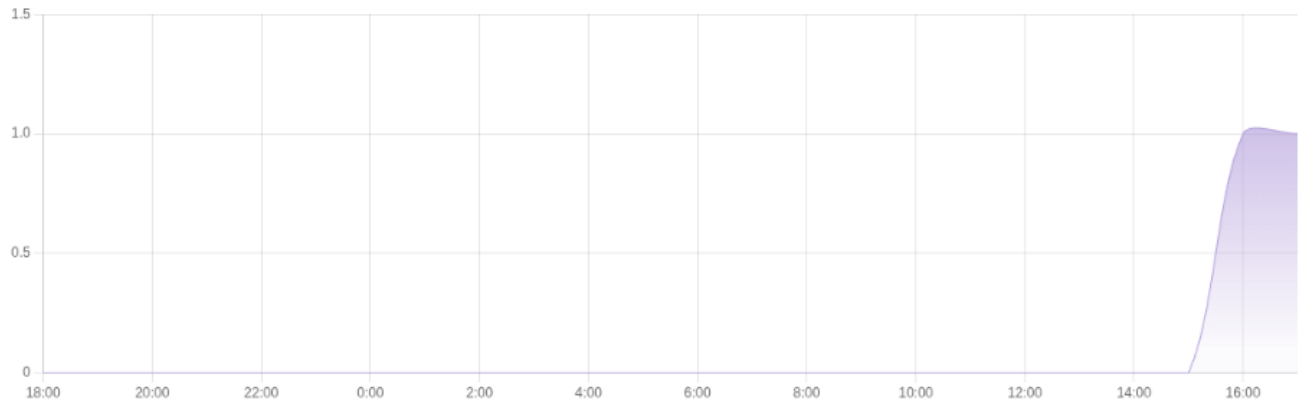
Simple view count, and if you click on the purple button, you get there :

Total views: **2**

Year Month Week Day

2 tracked visits in the last day.

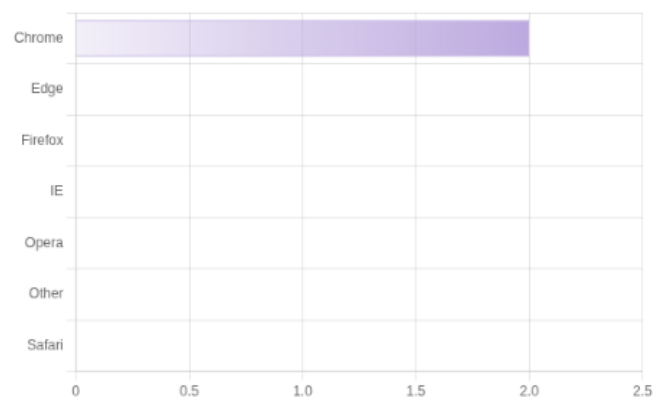
Last update at 17:54.



Referrers.



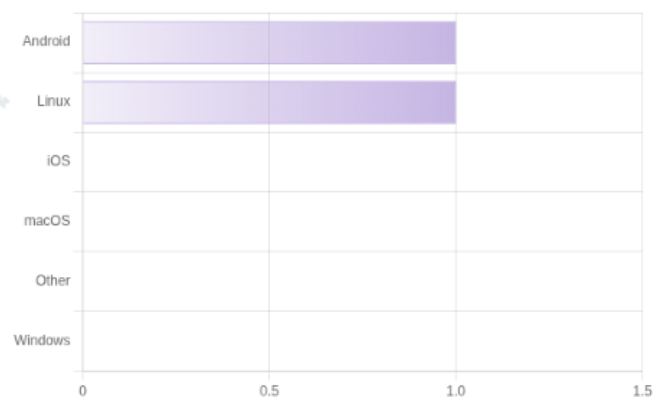
Browsers.



Countries.



Operating systems.



I'll do a umami blog post separately, but this was kutty, as Kwutty

[illegible]

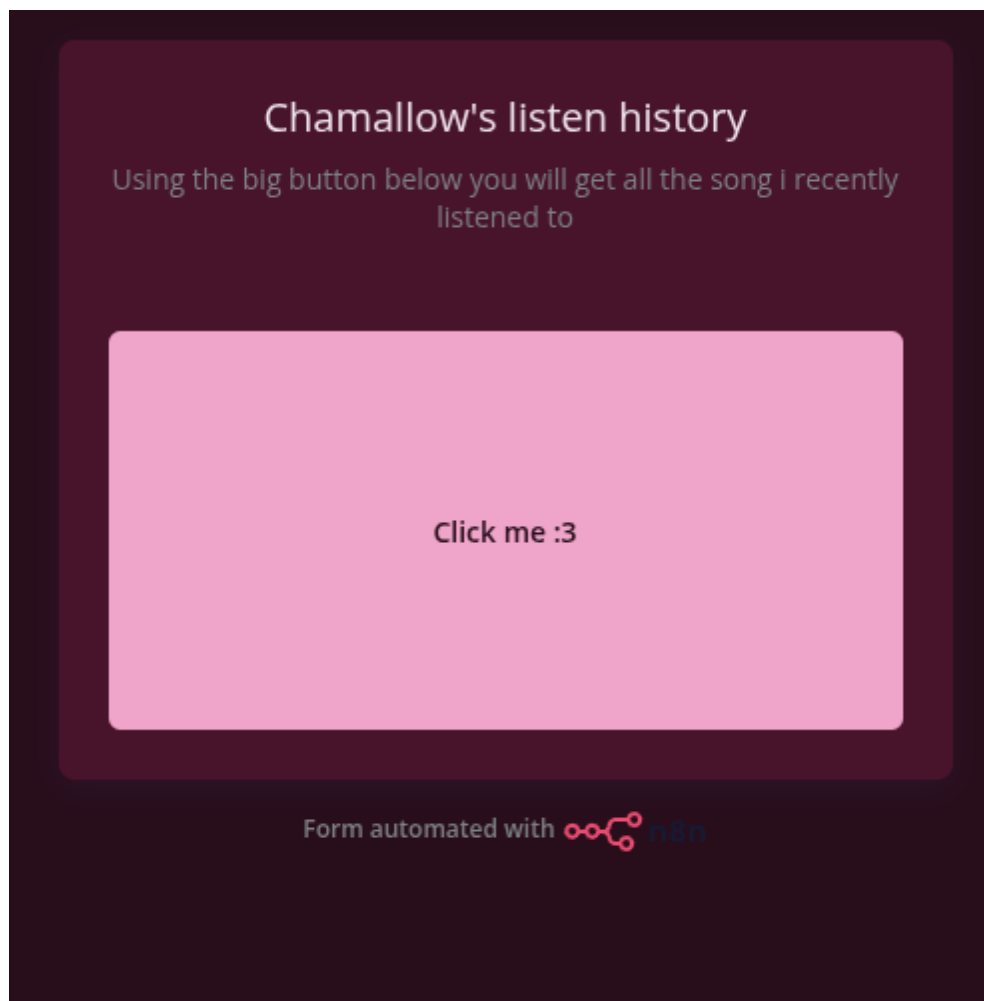
26-03-04: listening to banger song as i'm writing this

Bad code ahead, be warned !!
(please dont tell my mom about it)

n8n music history - [l.frezlotte.ch](https://n8n.frezlotte.ch)

http redirect to <https://n8n.frezlotte.ch/form/7766c9d4-aff1-452a-b455-a50d964793ac> but it's a bit much to remember i think :3

Landing page of the site, tf is that ?

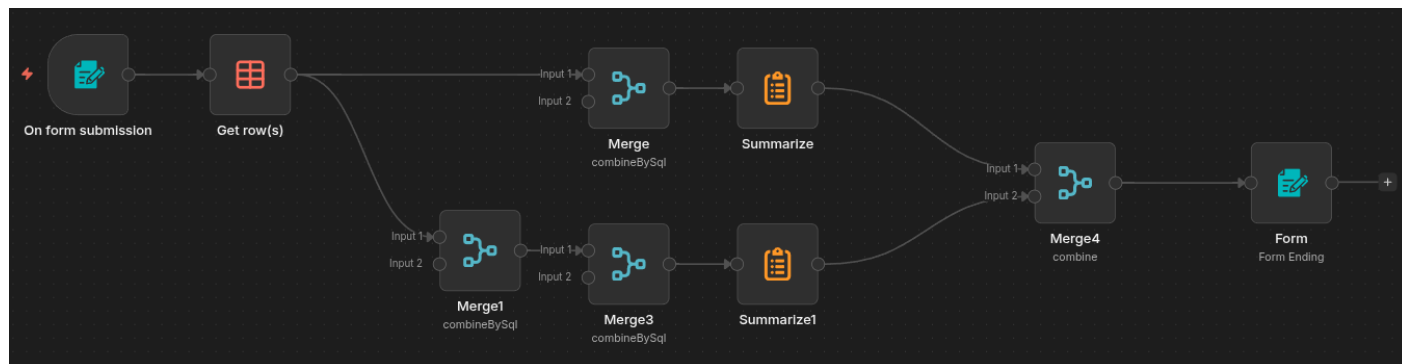


It's a n8n form, cause i don't want to do work constantly, pressing the button does the query and generate a static website

That's nor good nor useful but i thought it was funny, so i did

Website theme palette is <https://coolors.co/230614-460c28-f0a6ca-efc3e6-f0e6ef>

So what does it do exactly



Full form workflow

Form URLs

Test URL

Production URL

https://n8n.frezlotte.ch/form/7766c9d4-aff1-452a-b455-a50d964793ac

Authentication

None

Form Title

Chamallow's listen history

Form Description

Using the big button below you will get all the song i recently listened to

Form Elements

Currently no items exist

Add Form Element

On submission, the user will be taken to the next form node

Options

Button Label

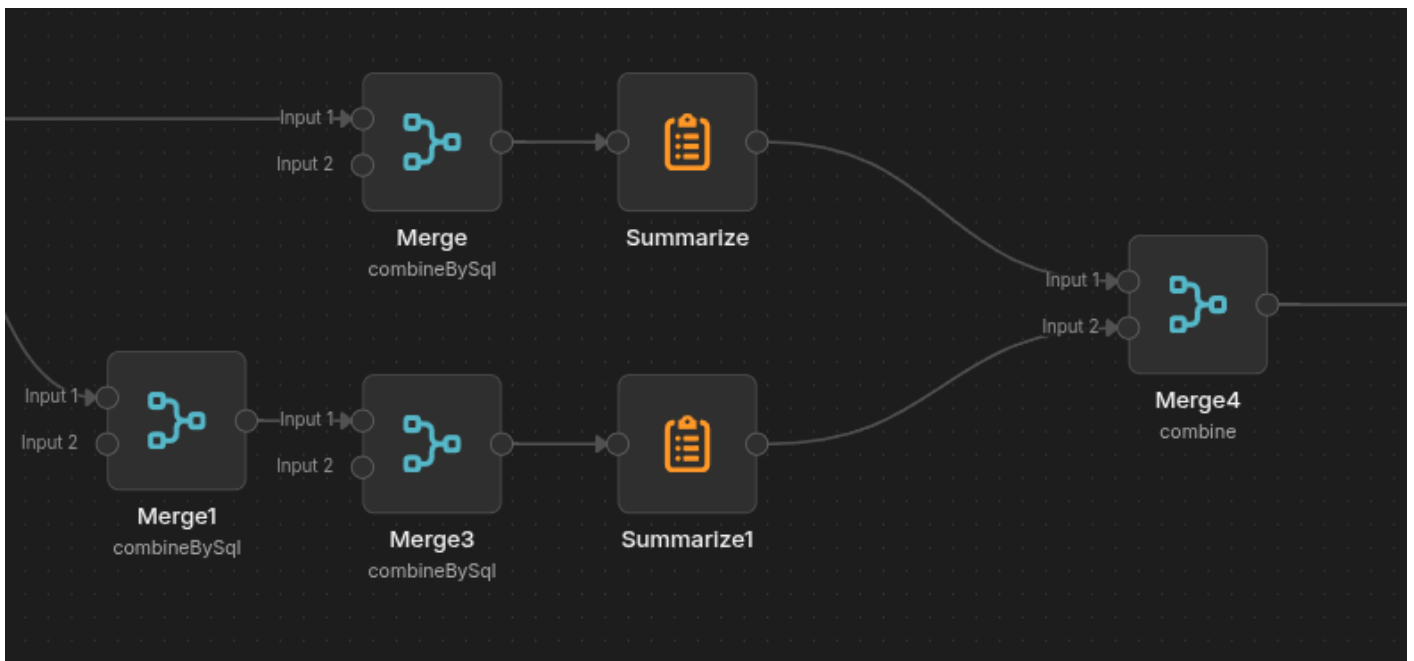
Click me :3

Custom Form Styling

```
1  √  :root {
2      --font-family: 'Open Sans', sans-serif;
3      font-weight: normal; 100%;
```

Simple form setup from n8n, Title description and some custom css and that's it

Then we get rows from a table we created manually before, settings are straight forward, we want all records and order them by id



SQL queries to modify stuff, it's ugly and the "kinda" SQL-Lite sql-99 "almost" pisses me the f off but at least it works now:

32 last song listened to

```
SELECT
  concat('<a href="',
        db.ShareUrl,
        '?u" target="_blank"><div class="musicItem" style="background-image: url(',
        db.SongArt,
        ')"><p>',
        db.SongArtiste,
        '</p><p>',
        db.SongName,
        '</p>',
        '<p><span class="iconify" data-icon="',
        IF(db.IsLiked, "mdi-heart", "mdi-heart-outline"),
        '"></span></p>',
        '</div>',
        '</a>'
  ) AS HTML_TEXT
FROM input1 as db
ORDER BY db.id desc
LIMIT 32
;
```

Table at the bottom of the page - count

```
SELECT
    SongArtiste AS Artiste,
    COUNT(*) AS ArtisteOccurrence,
    COUNT(DISTINCT SongName) AS UniqueMusicCount
FROM input1
GROUP BY SongArtiste
ORDER BY ArtisteOccurrence DESC
```

Table to HTML

```
SELECT
    CONCAT(
        '<tr>',
        '<td>',
        db.Artiste,
        '</td>',
        '<td>',
        '</td>',
        '<td>',
        db.ArtisteOccurrence,
        '</td>',
        '<td>',
        db.UniqueMusicCount,
        '</td>',
        '<td>',
        MID(db.MostListenSong, 1, 75),
        IF(LEN(MID(db.MostListenSong, 1, 75)) > 70, "....", ""),
        '</td>',
        '</tr>'
    ) as HTML_Code
FROM input1 as db
limit 50
;
```

Then we use the SUMMARIZE not to take each line and append them in one single string

we merge both output in one JSON object so we can use it at the same time to do the last step

HTML page generation

Parameters

Settings

Page Type

Form Ending

On n8n Form Submission

Show Text

Text

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Cha listen history</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

with n8n's form we can redirect to a site, or just send back raw HTML

Full template code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Cha listen history</title>
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <meta name="theme-color" content="#f06e42">
  <script src="https://code.iconify.design/1/1.0.6/iconify.min.js"></script>
<style>
@import
url('https://fonts.googleapis.com/css2?family=Alexandria:wght@100..900&display=swap');
</style>

<style>
body, html {
  font-family: Alexandria;
```

```
margin: 0;
background: #230614;
overflow-x: hidden;
}

body {
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}

body * {
  transition: .9s ease-in-out;
}

a:link {
  color: unset;
  text-decoration: none;
}

a:visited {
  color: unset;
  text-decoration: none;
}

a:hover {
  color: unset;
  text-decoration: none;
}

a:active {
  color: unset;
  text-decoration: none;
}

.musicItem svg {
  color: #ffc3c3 !important;
```

```
    filter: unset !important;
}
```

```
.musicItem {
  background: #EFC3E6;
  border-radius: 300px;
  width: 300px;
  height: 300px;
  margin: 25px;
  background-position: center;
  background-size: cover;
  background-repeat: no-repeat;
  display: flex;
  justify-content: end;
  flex-direction: column;
  padding: 5px 5px 10px 5px;
  box-shadow: 2px 2px 22px 22px #13020480,
              0px 15px 30px 10px #460C28f0 inset,
              0px -60px 50px 5px #460C28f0 inset,
              0px -60px 50px 5px #0000000f inset,
              0px 60px 110px 3px #ffffff55 inset;
  transition: .9s ease-in-out;
}
```

```
body:has(.musicItem:hover) *:not(:hover) {
  filter: saturate(0.5) brightness(0.9);
}
```

```
.musicContainer:has(.musicItem:hover) .musicItem:not(:hover) {
}
```

```
.musicItem:hover {
  transform: scale(1.2);
  box-shadow: -4px 6px 20px 10px #130204cf,
              0px 15px 30px 1px #460C28f0 inset,
              0px -60px 50px 5px #460C28f0 inset,
              0px 60px 110px 3px #ffffff55 inset,
              20px -40px 10px 3px #00000066 inset,
              20px -40px 60px 3px #ffffff55;
```

```
    filter: saturate(1.75);
}

.musicItem:hover p {
    transition: .75s ease-in-out;
    transform: scaleX(1.4) scaleY(1.4) translateY(-10px);
    margin: 10px auto;
    color: #F0E6EF;
    text-shadow: -1px -1px 1px #000, 1px -1px 1px #000,
                -1px 1px 1px #000, 1px 1px 1px #000,
                0px 4px 5px #000000;
}

.musicItem:hover p:nth-child(odd) {
    font-weight: 700;
    filter: saturate(2);
}

.musicItem:hover p:nth-child(even) {
    font-size: 1.2em;
    font-weight: 700;
    filter: saturate(2);
}

.musicItem p {
    color: #E1b8d6d9;
    padding: 0 0px;
    margin: 0 auto;
    text-align: center;
    max-width: 250px;
    text-shadow: 2px 4px 5px #130204,
                -5px 0 4px #130204;
    transition: .4s ease;
}

.musicItem p:nth-child(odd) {
    font-size: 2em;
    font-weight: 700;
}

.musicItem p:nth-child(even) {
```

```
font-size: 1.2em;
font-weight: 500;
}
```

```
.musicItem p:last-child {
    border-radius: 0 0 25px 25px;
}
```

```
.musicContainer {
    display: flex;
    justify-content: space-evenly;
    flex-wrap: wrap;
    flex-direction: row;
    width: 100vw;
    align-items: center;
    height: 100%;
    transition: .75s ease-in-out;
    margin: auto;
    max-width: 1500px;
}
```

```
h1 {
    text-align: center;
    font-size: 3.4em;
    width: 100vw;
    margin: 0;
    padding: 150px 20px 0px 20px;
    color: #EFC3E6;
    text-shadow: 0px 0px 2px;
    background: linear-gradient(#130204ff, #13020400);
}
```

```
h4 {
    text-align: center;
    font-size: 1.2em;
    font-weight: 300;
    width: 100vw;
    margin: 0;
```

```
padding: 0px 0px 0px 0px;
color: #EFC3E6;
text-shadow: 0px 0px 2px;
}
```

```
h3 {
  text-align: center;
  font-size: 1.2em;
  font-weight: 300;
  width: 100vw;
  margin: 0;
  padding: 0px 0px 150px 0px;
  color: #EFC3E6;
  text-shadow: 0px 0px 2px;
}
```

```
footer {
  height: 150px;
  background: linear-gradient(#13020400, #130204ff);
  color: #F0A6CA;
}
```

```
table {
  border: 1px solid #460C28;
  margin: 50px auto;
  padding: 20px;
  color: #F0E6EF;
  border-radius: 10px;
  box-shadow: 2px 2px 22px 22px #13020480;
  font-size: 1.3em;
  max-width: 1500px;
  th {
    text-align: left;
  }
  td, th {
    padding: 5px 5px 5px 5px;
  }
  tr:nth-child(odd) {
    background-color: #460C28;
```



```

    }
    tr:nth-child(even) {
        background-color: #230614;
    }
}
</style>
</head>
<body>
    <a href="http://l.frezlotte.ch">
        <h1>Chamallow's listening history <span class="iconify" data-icon="mdi-heart"></span></h1>
        <h4>click up here to refresh the page</h4>
        <h3><span class="iconify" data-icon="mdi-refresh"></span></h3>
    </a>
    <div class="musicContainer">
        {{ $json.concatenated_HTML_TEXT }}
    </div>
    <table>
    <tr>
        <th>Artiste</th>
        <th>listen</th>
        <th>Total</th>
        <th>Unique</th>
        <th>Most listened Title</th>
    </tr>
    {{ $json.concatenated_HTML_Code }}
    </table>
    <footer>
<p style="width: 100%; text-align: center; transform: translateY(50px)">
    Made with love <a style="text-decoration: underline;" href="https://frezlotte.ch">by
    myself</a>
</p>
    </footer>
</body>
</html>

```

important stuff are :

```

{{ $json.concatenated_HTML_TEXT }}

```

```
{{ $json.concatenated_HTML_Code }}
```

My names are shit but HTLM_TEXT (typo ik lmao) does the history on the page, and the HTML_Code is for the table

Can't bother

Feeding the table with songs akshtually

Webhooks

Like i did before with minecraft to have a 2 way chat system, i used webhooks, n8n is good for that you can easily setup data and have some ipAllowList so that not everyone can just POST on it (or you can also use diff kind of Auths, pretty based)



Resource

Row

Operation

Insert

Data table

By Name

ListenedTo

Mapping Column Mode

Map Each Column Manually

Values to Insert

ShareUrl

fx

{{ \$json.body.share_url }}

SongName

fx

{{ \$json.body.song_name }}

SongArtiste

fx

{{ \$json.body.song_artiste }}

IsLiked

fx

{{ \$json.body.is_liked }}

SongArt

fx

{{ \$json.body.song_art }}

Options

Very simple stuff

Automated POST requests for TIDAL

i started from <https://github.com/titaniumfish/tidal-discord-richpresence>

It missed a lot of features tho, and i really wanted to display images, tho i had an issue

On Tidal, music can have some NSFW artwork, and i knew that wouldn't be okay with discord...

So i added A module to call Tidal API, to see songs metadata i'm missing, and as a middle ground i show the album cover in the small icon using the "/icon" endpoint so 80x80, should be okay

-> tho i saw since then that some song have NSFW artwork but no E rating, that's annoying but not much i can do about it

Am putting the code below, it's bad and i vibe coded some of it but it works :3

cya [REDACTED]

```
#!/usr/bin/env python3

import json
import logging
import platform
import re
import subprocess
import time
from pathlib import Path
from typing import Any, Dict, Optional

import requests
from pypresence import Presence
from pypresence.types import ActivityType, StatusDisplayType
from tidalapi import Session
from tidalapi.media import Track

DISCORD_CLIENT_ID = "1477481718729674793"
TIDAL_API_BASE = "http://localhost:47836"
TIDAL_CURRENT_ENDPOINT = f"{TIDAL_API_BASE}/current"
TIDAL_CURRENT_IMAGE = f"{TIDAL_API_BASE}/current/image"
UPDATE_INTERVAL = 5 # seconds between updates
CONNECT_RETRY_INTERVAL = 30 # seconds between Discord connection attempts

def report_to_n8n(share_url, song_name, song_artiste, is_liked, song_art):
    payload = {
        "share_url": share_url,
        "song_name": song_name,
        "song_artiste": song_artiste,
        "is_liked": is_liked,
        "song_art": song_art,
    }
    try:
```

```

        response = requests.post(
            "https://user:pswr@n8n.frezlotte.ch/webhook/ad127f87-cbae-4fde-800b-23b6e3f6a6f6",
            json=payload,
            timeout=5,
        )
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        pass

```

```

class TidalDiscordRPC:
    def __init__(self, tidal_api):
        self.discord_rpc = None
        self.last_track_data = None
        self.connected_to_discord = False
        self.api = tidal_api

        logging.basicConfig(
            level=logging.INFO, format="%(asctime)s - %(levelname)s - %(message)s"
        )
        self.logger = logging.getLogger(__name__)
        # logging.getLogger().setLevel(logging.DEBUG)

    def connect_discord(self) -> bool:
        try:
            if self.discord_rpc:
                self.discord_rpc.close()

            self.discord_rpc = Presence(DISCORD_CLIENT_ID)
            self.discord_rpc.connect()
            self.connected_to_discord = True
            self.logger.info("Connected to Discord Rich Presence")
            return True

        except Exception as e:
            self.logger.error(f"Failed to connect to Discord: {e}")
            self.connected_to_discord = False
            return False

```

```

def get_current_track(self) -> Optional[Dict[str, Any]]:
    try:
        response = requests.get(TIDAL_CURRENT_ENDPOINT, timeout=5)
        response.raise_for_status()

        data = response.json()

        if data.get("title") and data.get("artists"):
            return data
        else:
            return None

    except requests.exceptions.RequestException as e:
        self.logger.error(f"Failed to get track info from Tidal Hi-Fi: {e}")
        return None
    except json.JSONDecodeError as e:
        self.logger.error(f"Invalid JSON response from Tidal Hi-Fi: {e}")
        return None

def format_time(self, seconds: int) -> str:
    minutes = seconds // 60
    seconds = seconds % 60
    return f"{minutes}:{seconds:02d}"

def update_discord_presence(self, track_data: Dict[str, Any]) -> bool:
    if not self.connected_to_discord:
        return False
    try:
        title = track_data.get("title", "Unknown Track")
        url = track_data.get("url", "")
        track_id_match = re.search(r"/track/(\d+)", url)
        track_id = track_id_match.group(1) if track_id_match else None
        track_api_data: "Track" = self.api.track(track_id)
        share_url = track_api_data.share_url
        image = track_data.get("image", "tidal-logo")
        artists = track_data.get("artists", "Unknown Artist")
        status = track_data.get("status", "unknown")
        current_seconds = track_data.get("currentInSeconds", 0)
        duration_seconds = track_data.get("durationInSeconds", 0)
        is_explicit = track_api_data.explicit or False

```

```

small_image = "play" if status == "playing" else "pause"

if is_explicit:
    small_image = image
    image = "https://cdn-icons-png.flaticon.com/512/8035/8035031.png"

bpm = track_api_data.bpm
popu = track_api_data.popularity
audio_qual = track_api_data.audio_quality
is_liked = track_data.get("favorite")
spacer = " - "
liked_char = "❤️"
description = liked_char if is_liked else "♡?"
description += " "
description += f"{bpm}bpm" + spacer if bpm else ""
description += f"{popu}%" + spacer if popu else ""
description += f"{audio_qual}" if audio_qual else ""

state = liked_char + " " if is_liked else ""
state += "[E] " if is_explicit else ""
state += f"{artists}{spacer}{title}"

presence_data = {
    "activity_type": ActivityType.LISTENING,
    "state": state,
    "large_text": description,
    "status_display_type": StatusDisplayType.STATE,
    "buttons": [{"label": "url", "url": share_url}],
    "large_image": image,
    "small_image": small_image,
    "small_text": "Playing" if status == "playing" else "Paused",
}

if status == "playing" and duration_seconds > 0:
    current_time = time.time()
    start_time = current_time - current_seconds
    end_time = start_time + duration_seconds

    presence_data["start"] = int(start_time)
    presence_data["end"] = int(end_time)

```

```

self.discord_rpc.update(**presence_data)
self.logger.debug(f"Sent presence data: {presence_data}")
report_to_n8n(share_url, title, artists, is_liked, track_data.get("image"))
time_info = ""
if current_seconds and duration_seconds:
    time_info = f"

[{self.format_time(current_seconds)}/{self.format_time(duration_seconds)}]"

self.logger.info(
    f"Updated Discord: {title} by {artists} ({status}){time_info}"
)
return True

except Exception as e:
    self.logger.error(f"Failed to update Discord presence: {e}")
    self.logger.error(
        f"Presence data that failed: {presence_data if 'presence_data' in locals()
else 'N/A'}"
    )
    self.connected_to_discord = False
    return False

def clear_discord_presence(self):
    if self.connected_to_discord and self.discord_rpc:
        try:
            self.discord_rpc.clear()
            self.logger.info("Cleared Discord presence")
        except Exception as e:
            self.logger.error(f"Failed to clear Discord presence: {e}")

def has_track_changed(self, current_data: Dict[str, Any]) -> bool:
    if not self.last_track_data:
        return True

    current_key = (
        current_data.get("favorite"),
        current_data.get("title"),
        current_data.get("artists"),
        current_data.get("status"),

```



```

)

last_key = (
    self.last_track_data.get("favorite"),
    self.last_track_data.get("title"),
    self.last_track_data.get("artists"),
    self.last_track_data.get("status"),
)

return current_key != last_key

def run(self):
    self.logger.info("Starting Tidal Hi-Fi Discord Rich Presence...")
    last_discord_attempt = 0

    while True:
        try:
            current_time = time.time()
            if (
                not self.connected_to_discord
                and (current_time - last_discord_attempt) > CONNECT_RETRY_INTERVAL
            ):
                self.connect_discord()
                last_discord_attempt = current_time
            track_data = self.get_current_track()

            if track_data and track_data.get("status") == "playing":
                if self.has_track_changed(track_data):
                    if self.connected_to_discord:
                        success = self.update_discord_presence(track_data)
                        if not success:
                            self.connected_to_discord = False

                    self.last_track_data = track_data
                else:
                    if self.last_track_data and self.connected_to_discord:
                        self.clear_discord_presence()
                        self.last_track_data = None

            time.sleep(UPDATE_INTERVAL)

```

```

        except KeyboardInterrupt:
            self.logger.info("Shutting down...")
            if self.connected_to_discord:
                self.clear_discord_presence()
            break
        except Exception as e:
            self.logger.error(f"Unexpected error: {e}")
            time.sleep(UPDATE_INTERVAL)

    if self.discord_rpc:
        self.discord_rpc.close()

if __name__ == "__main__":
    # Wait until Discord (vesktop) and Tidal are running before proceeding
    while True:
        try:
            if platform.system() == "Windows":
                output = subprocess.check_output("tasklist", universal_newlines=True)
                discord_running = "Discord.exe" in output
                tidal_running = "Tidal.exe" in output
            else:
                output = subprocess.check_output(["ps", "aux"], universal_newlines=True)
                discord_running = "vesktop" in output
                tidal_running = "tidal-hifi" in output

            if discord_running and tidal_running:
                break
        except Exception:
            pass
        time.sleep(5)

    session_file1 = Path("tidal-session-oauthn.json")
    session = Session()
    session.login_session_file(session_file1)
    rpc = TidalDiscordRPC(session)
    rpc.run()

```

26-02-23: Waaaa, s t u f f

AWO

I changed ☐ stuff ☐ on the wiki

now i'm gonna do blogs and research stuff on here

gonna be so cooooooooool

- Cha



25-07-16: me like music and cool visuals

Heres 2 visuals i'm starting to be proud of, am really new to VVVV but i'm addicted already

!!! Photosensible warning --- Flashing lights ahead !!!

Technical details;

I'm using my audient evo 4 - loopback, that i monitor to my headphones so i can listen at the same time, very useful i'm glad i didn't had to use virtual audio cable.

Also tried to wire some "intruments" but it had random effects, cutting my mic, cutting my sound, sending the input back into my hears for some reason... I'll try again someday but for now it's fine with the loopback technique.

Nodes and stuff

if y'all want to see how it's done let me know, am not gonna write shit no one gonna read

[Music: Fireflies - Al'Tarba](#)

[Music: THINGS WILL GET MUCH WORSE FROM HERE - Kill Bill - Rav](#)

and after a few more hours ...