lrovaris

CODE MANAGEMENT

non fungible bubz

technical code guide 2021

TABLE OF CONTENTS

01 Running Cardano

02 Setup a drop

03 Work with Policy IDs

04 Do random free airdrops on wallets

05 Setup NFT fusion

06 Setup action for fusion

07 Add new projects to the the data dashboard

RUNNING CARDANO

LRovaris

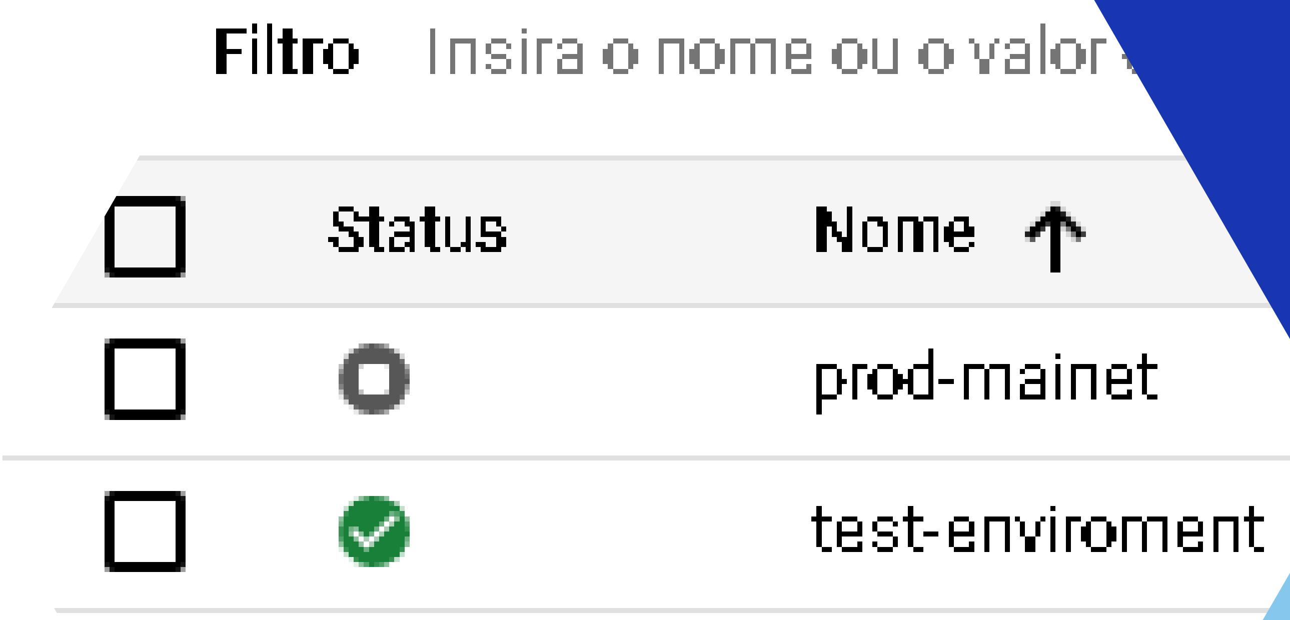
```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;


robabilities = [ 0.5, 0.3, 0.15
```

TECHNICAL CODE GUIDE 2021

WHERE DO I START?

How We Started

those are two strong questions you might see, therefore there's no need to doubt, everything at the moment runs on your google cloud, there's where the main bill comes from, there is two virtual machines, both with Cardano-node / cli installed and synced with the database.
starting then should be intuitive by google, see =>



Filtro Insira o nome ou o valor

□	Status	Nome ↑
□	●	prod-mainet
□	✓	test-enviroment

starting the server

google cloud > compute engine > vm

there's how to get on the screen from the last picutere, and where you should do the following =>

Click the three dots on the right corner of the grid-line, and select ‘start.’ this should start a enviroment

Click ‘ssh’ should open a new window with a linux bash (similar to your mac bash)

in order to get what i've done you got go to my user home dir, to do that: ‘r cd .. / ls -la (to get my user to copy, should be luis_gab...) / cd luis_gab...’

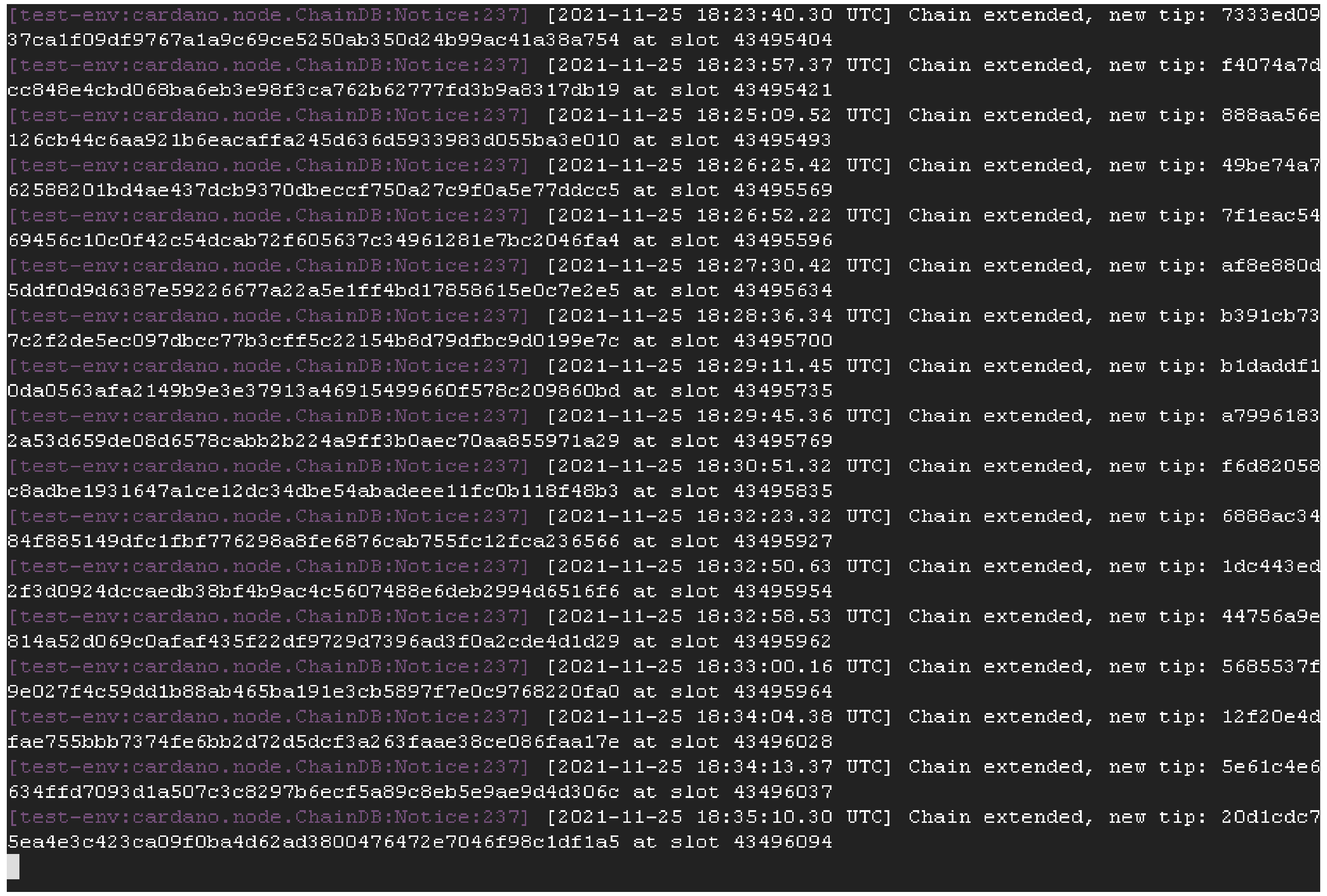
then input ‘cd VendingMachine/server’ in orther to navigate to the server folder.

input 'git pull' in order to get the last server code, note that it might require a password, or github auth token in order to git pull after you close the repo.

In order to prevent the server from shutdown after closing the window input: 'screen -S api' or any other screen name rather than api.

to start the test server run 'npm start test' or 'npm start prod' in order to run the production server

SYNC WITH THE BLOCKCHAIN



```
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:23:40.30 UTC] Chain extended, new tip: 7333ed09
37ca1f09df9767a1a9c69ce5250ab350d24b99ac41a38a754 at slot 43495404
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:23:57.37 UTC] Chain extended, new tip: f4074a7d
cc848e4cbd068ba6eb3e98f3ca762b62777fd3b9a8317db19 at slot 43495421
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:25:09.52 UTC] Chain extended, new tip: 888aa56e
126cb44c6aa921b6eacaffa245d636d5933983d055ba3e010 at slot 43495493
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:26:25.42 UTC] Chain extended, new tip: 49be74a7
62588201bd4ae437dcb9370dbeccf750a27c9f0a5e77ddcc5 at slot 43495569
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:26:52.22 UTC] Chain extended, new tip: 7f1eac54
69456c10c0f42c54dcab72f605637c34961281e7bc2046fa4 at slot 43495596
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:27:30.42 UTC] Chain extended, new tip: af8e880d
5ddf0d9d6387e59226677a22a5e1ff4bd17858615e0c7e2e5 at slot 43495634
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:28:36.34 UTC] Chain extended, new tip: b391cb73
7c2f2de5ec097dbcc77b3cff5c22154b8d79dfbc9d0199e7c at slot 43495700
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:29:11.45 UTC] Chain extended, new tip: b1daddf1
0da0563afa2149b9e3e37913a46915499660f578c209860bd at slot 43495735
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:29:45.36 UTC] Chain extended, new tip: a7996183
2a53d659de08d6578cabb2b224a9ff3b0aec70aa855971a29 at slot 43495769
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:30:51.32 UTC] Chain extended, new tip: f6d82058
c8adbe1931647a1ce12dc34dbe54abadeee11fc0b118f48b3 at slot 43495835
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:32:23.32 UTC] Chain extended, new tip: 6888ac34
84f885149dfc1fbf776298a8fe6876cab755fc12fca236566 at slot 43495927
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:32:50.63 UTC] Chain extended, new tip: 1dc443ed
2f3d0924dccaedb38bf4b9ac4c5607488e6deb2994d6516f6 at slot 43495954
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:32:58.53 UTC] Chain extended, new tip: 44756a9e
814a52d069c0afaf435f22df9729d7396ad3f0a2cde4d1d29 at slot 43495962
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:33:00.16 UTC] Chain extended, new tip: 5685537f
9e027f4c59dd1b88ab465ba191e3cb5897f7e0c9768220fa0 at slot 43495964
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:34:04.38 UTC] Chain extended, new tip: 12f20e4d
fae755bbb7374fe6bb2d72d5dcf3a263faae38ce086faa17e at slot 43496028
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:34:13.37 UTC] Chain extended, new tip: 5e61c4e6
634ffd7093d1a507c3c8297b6ecf5a89c8eb5e9ae9d4d306c at slot 43496037
[test-env:cardano.node.ChainDB:Notice:237] [2021-11-25 18:35:10.30 UTC] Chain extended, new tip: 20d1cdc7
5ea4e3c423ca09f0ba4d62ad3800476472e7046f98c1df1a5 at slot 43496094
```

if successful you should see the "chain extended" message...
but it takes a while, sometimes even an entire hour...

Easy as one command

Yyou can visit: https://developers.cardano.org/docs/get-started/running-cardano for more details about what you're going to copy

```
cardano-node run \
--topology testnet-topology.json \
--database-path ./db \
--socket-path ./db/node.socket \
--host-addr 10.128.0.3 \
--port 3001 \
--config testnet-config.json
```

note the "testnet- before some paths... in order to run the prod blockchain you just got replace testnet- for mainnet-

as well the ip address of the VM

USING CARDANO-CLI

the server is using cardanoCli-js it's a npm library to run cardano-cli bash commands with javascript, you can look their index on node_modules folder to understand better what's possible to do with it

running scripts

running scripts that cosumes cardanoCli-js module requires first to import a enviroment variable called CARDANO_NODE_SOCKET_PATH

socket path

so where do we get the node socket? if we try to run any code at this moment the bash ill return error.
note the last code we copied in order to run the cardano blockchain?
note a path there for node.socket file

Copy

export CARDANO_NODE_SOCKET_PATH=../../cardano-src/config/db/node.socket

if you are in VendingMachine/server

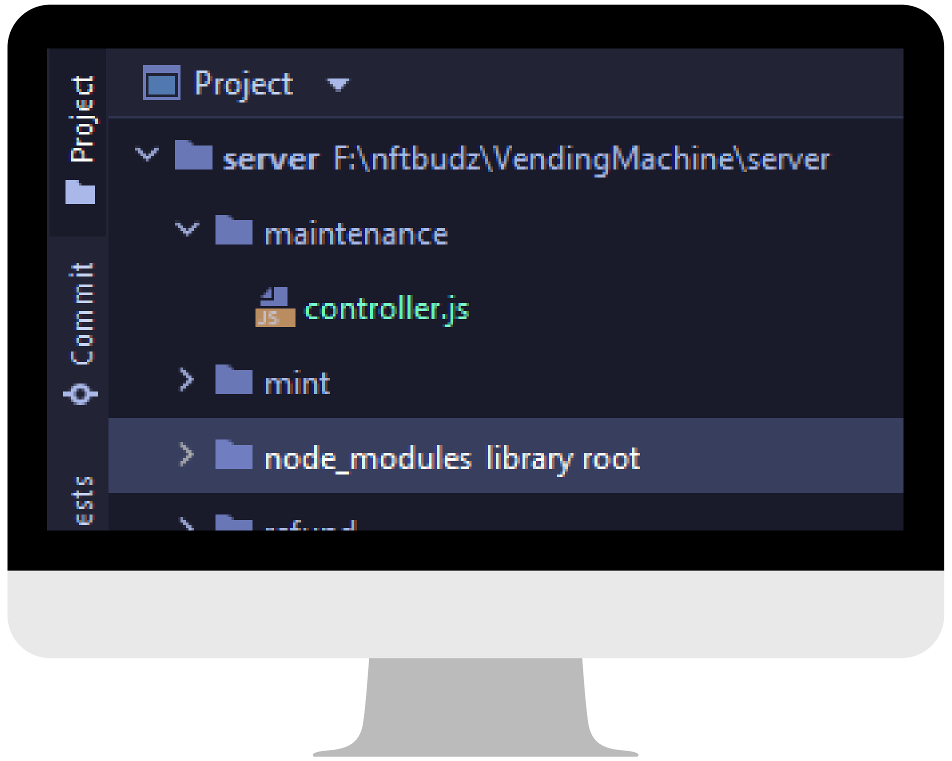
otherwise just replace the ../ regarding the folder you're from luis_gabrielrovaris home folder

SETTING UP A DROP

```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;

robabilities = [ 0.5, 0.3, 0.15
```

IMPORTANT THINGS

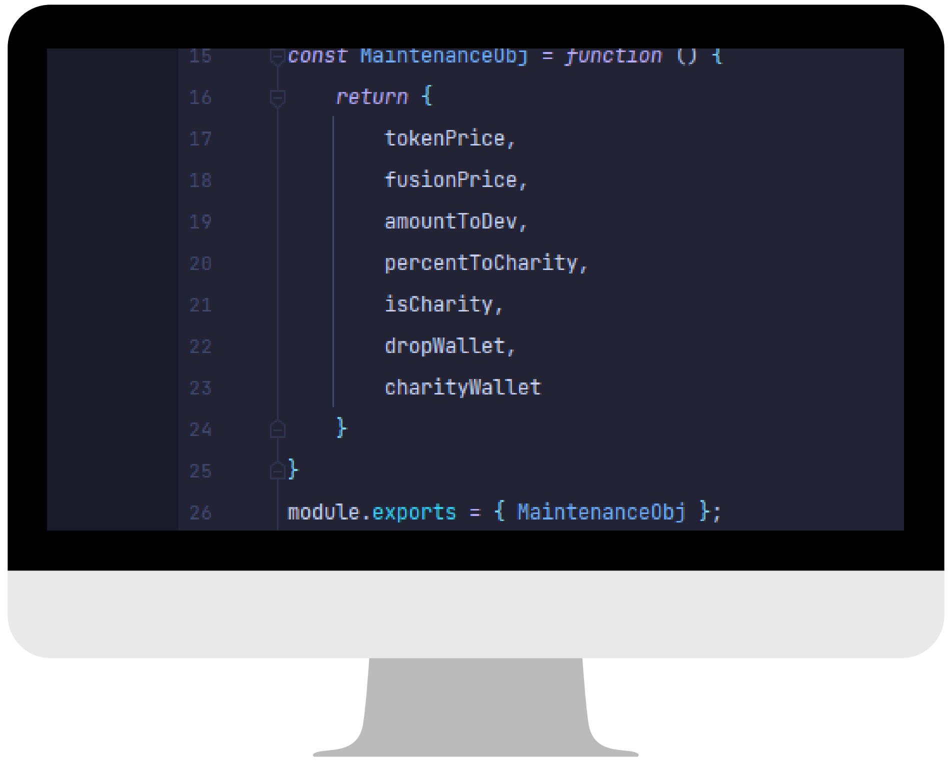
In order to edit things in a easy way, i've made a script...



```
server  F:\nftbudz\VendingMachine\server
    maintenance
        controller.js
    mint
    node_modules  library root
```

server/maintenance

controller.js ->
all the important things i've put in
a single script, with only that, so
there's no mistery relating to edit
those values



```
const MaintenanceObj = function () {
    return {
        tokenPrice,
        fusionPrice,
        amountToDev,
        percentToCharity,
        isCharity,
        dropWallet,
        charityWallet
    }
}
module.exports = { MaintenanceObj };
```

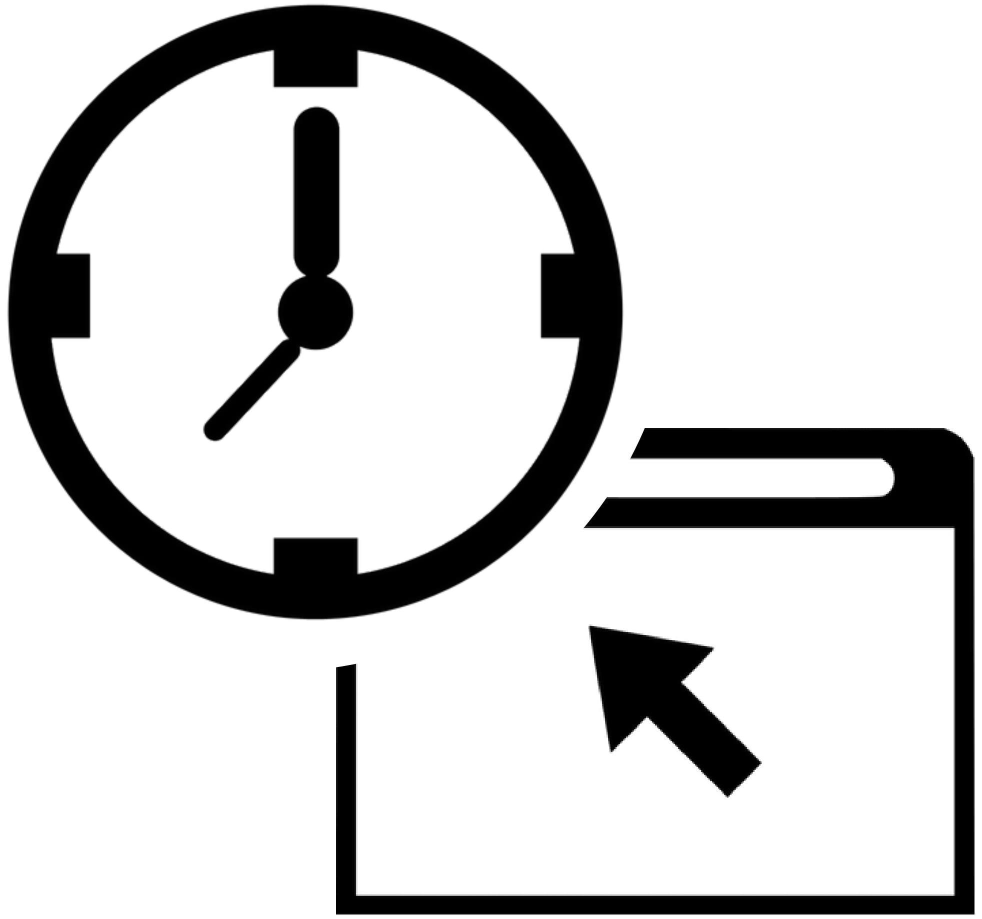
object returned

the script just returns a single
object and it's used in
mint/controller/automint.js

that concludes

setting up price for drop
% for charity
amount to dev
price to fusion
and the wallets to receive the
amount, dev wallet isn't there

CRONJOBS

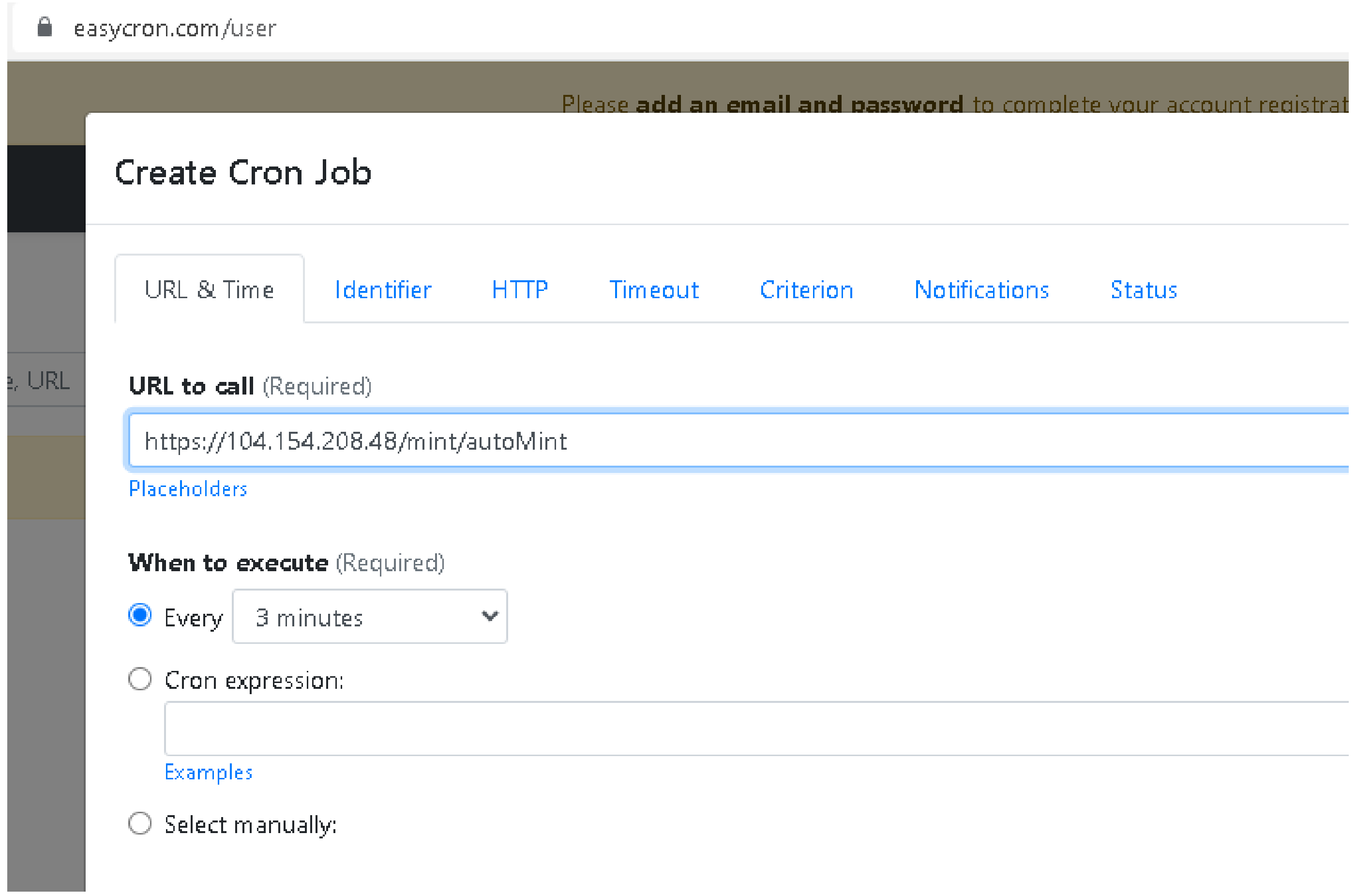


repetitive tasks

our mint/fuse main functionality

basically i did things the server in a way that all we need is a bot that call our mint/fuse handlers once in a while... so I've opted to do that with cronjobs to minimize cost, cronjobs can be easily setup, i did one for mint handler and another for fuse handler, few cronjobs should be free anywhere that works with it (i think there are even possible to do in the linux vm, but that may be way too much work), therefore it can be easily found online.

THE JOBS



Please **add an email and password** to complete your account registrat

Create Cron Job

URL to call (Required)

https://104.154.208.48/mint/autoMint

Placeholders

When to execute (Required)

● Every 3 minutes

○ Cron expression:

Examples

○ Select manually:

urls and time between calls

the only two things you got worry when doing cronjobs like that, note that in this printscreen it gots the external ip from the test vm (and it also changes every time you restart the vm, so don't worry about that too much)

Either is the time between calls so needy of worries, it's mandatory of course, that the routes get called in order the service both mint and fuse works, if it's not calling then its not working... but time maybe more like a personal issue, but considerer a minimum time of 3 minutes, because cardano-cli has a bit of a lag after making a transaction, something between 1 and 2 minutes, so that's why i've put the cron job with 3 minutes

CONCLUSION SET UP A DROP

now that we now what cronjobs are, and a little on how the system is working, by the time you're reading this guide, your system is up and running and is actually possible to mint a bub just by sending 25 ada to the mint wallet, after 3 to 6 minutes you should receive a nft bub just by sending ada to the server wallet.

SO, IN ORDER TO SET UP, OR PAUSE A DROP

There are two main things to turn on/off in order to work or not, first one is the server, earlier in this guide you've read on how to reach the server folder, there you should type 'npm start prod' . for the actual drop or 'npm start test' in the test environment

The second one are turning on/off the cronjobs (you shouldn't turn then off actually, they're free... and you can just turn on/off by running or not the server). Therefore if the server is running and the cronjobs aren't your mint / fuse aren't working... but you can activate then manually, by calling the 'server.ip/mint/automint' or '/fuse', calling then twice should trigger the mint/fuse/refund function. You can use postMan for calling manually

WORKING WITH POLICYID

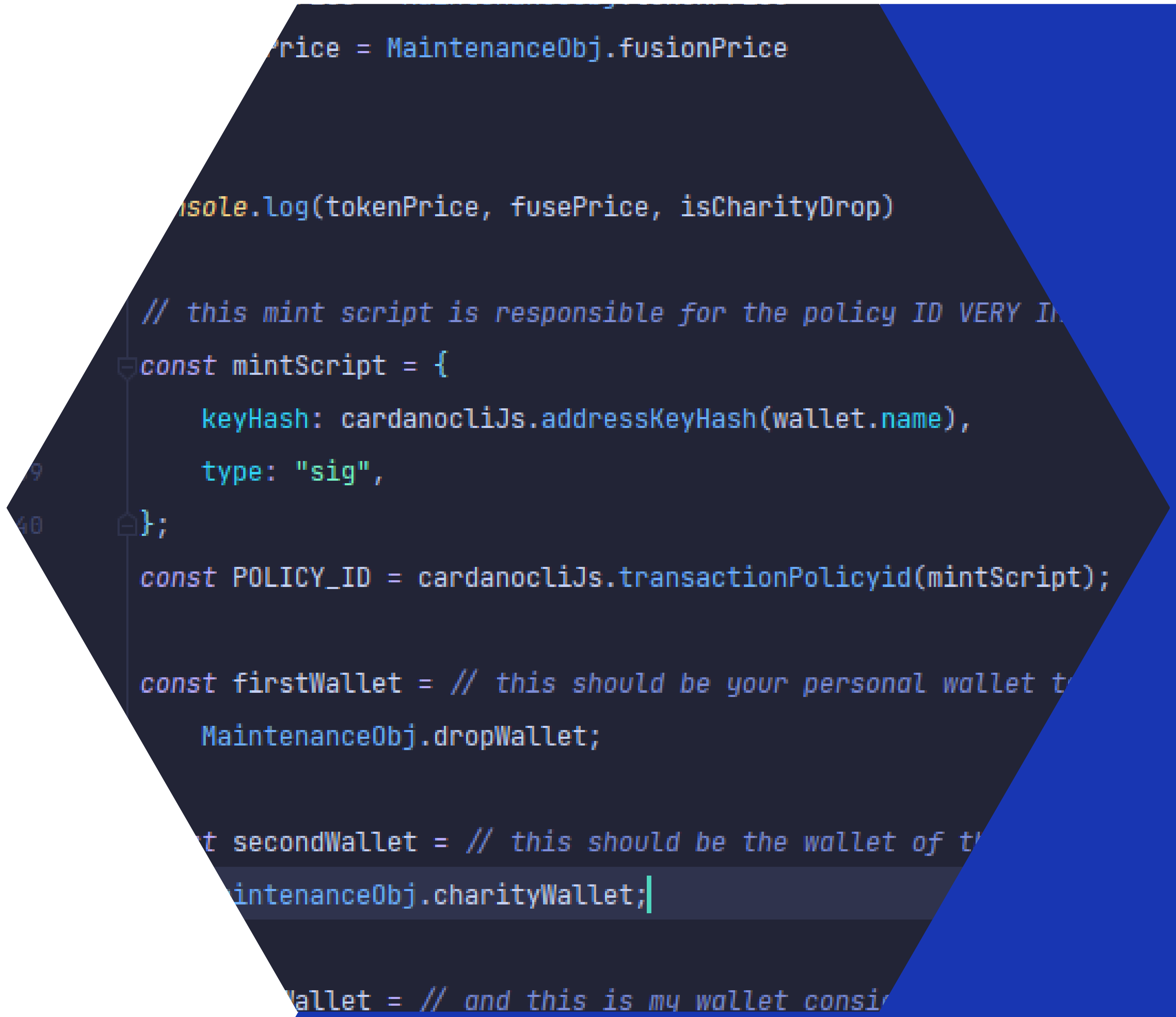
```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;

robabilities = [ 0.5, 0.3, 0.15
```

NO MISTERY

PolicyID = Wallet ID

Every wallet has signing keys inside cardano ecosystem, a different policyID means a different signing key, digging trough the code searching for the mintscript, and the policyID variables in the mint function might clarify what i mean...

tld dr : coming with a new policyID requires creating a new wallet, ive made a script for it just need to rename the wallet name in the code, and type in the linux vm inside the server folder 'node create-wallet.js prod' (or test)'



```
rice = MaintenanceObj.fusionPrice


sole.log(tokenPrice, fusePrice, isCharityDrop)

// this mint script is responsible for the policy ID VERY I
const mintScript = {
    keyHash: cardanocliJs.addressKeyHash(wallet.name),
    type: "sig",
};
const POLICY_ID = cardanocliJs.transactionPolicyid(mintScript);

const firstWallet = // this should be your personal wallet t
    MaintenanceObj.dropWallet;

t secondWallet = // this should be the wallet of t
intenanceObj.charityWallet;


allet = // and this is my wallet consi
```

LRovaris

UNFORTUNATELY

There's no easy way to see the wallet policyID in the code
But we can always see the policyID in a token that's been minted by that wallet
so, you check in your wallet, or pool.pm

RANDON FREE AIRDROPS

```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;

robabilities = [ 0.5, 0.3, 0.15
```

BASICALLY

in don't know how you do pretend to be in touch with your community, i adivice discord, it's a great platform there are a lot of suportive cardano servers, so baiscally the best way to do "randon" drops right now are not programatically, i can therefore manage to code this for you, if you like (if you're reading this out of 2021 this shouldn't be important, i might come with a new version of the doc anyway)

SO...

you can do the random in any website that does that for free, inputing the info from your community, or simply looking trough the wallet holders in cardanoscan.io
Ok... but how can i mint then a token? if you want to mint sometoken for "free" you can :3, they're not actually free, because you pay the fee from cardano network (around 200000 lovelace 0.2 ada per transaction, plus 1 to dev and 1.5 to the client, being send from the wallet that is in the system sending 25 ada in order to mint should return a token plus 23.4 ada 😎

THEREFORE

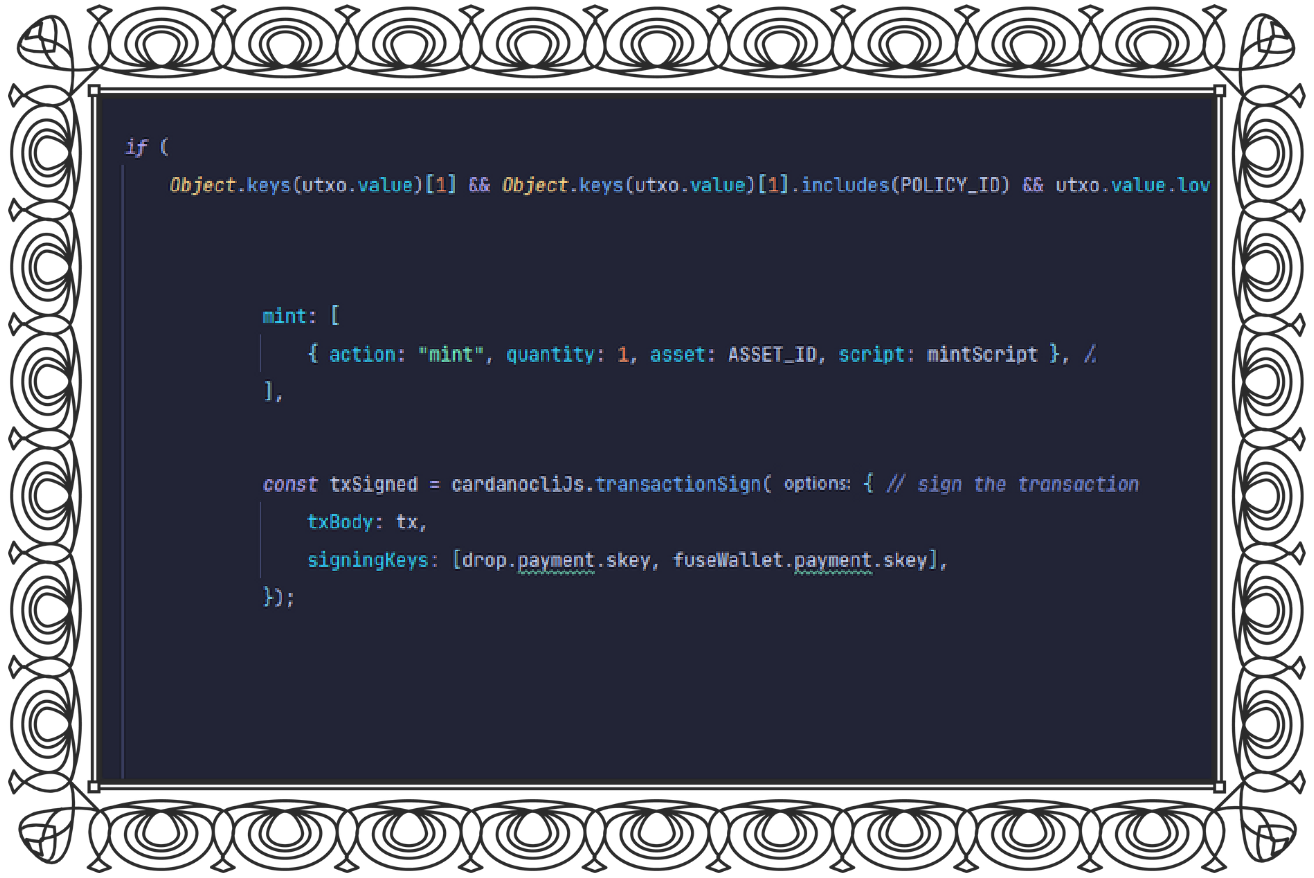
it's possible tho... to make it programatically, i'm using right now blockfrost to track txhash and make the automint / refund / fuse possibly , i can study it a little more and discorver how to track the policy id token holders of some kind and make drops totally randon programatically, but would cost at least 1.5 ada (or 2.5 if u consider the dev share xD)

NFT FUSION

```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;

robabilities = [ 0.5, 0.3, 0.15
```

DIFFERENT WALLET

let's recap real quick how's our system working right now... you send ada to one wallet, it takes 3 - 6 minutes (cronjob time) you get a token plus 1.5 ada, real nice huh? But how's fuse working? You might ask, is not that complex actually, works like mint, but also asks for a token with the policyID, why it got a different wallet from the mint? that means a different policty id right? Not quite... i've build the transaction script in a way that the "dropWallet" signs the mint and the fuse sign the transaction... confuse? a little xD check out, dual sign keys,



```
if (
    Object.keys(utxo.value)[1] && Object.keys(utxo.value)[1].includes(POLICY_ID) && utxo.value.lov



          mint: [
              { action: "mint", quantity: 1, asset: ASSET_ID, script: mintScript }, /
          ],


          const txSigned = cardanocliJs.transactionSign( options: { // sign the transaction
              txBody: tx,
              signingKeys: [drop.payment.skey, fuseWallet.payment.skey],
          });
```

know issues

you might face errors and can expect server shut down if someone send an token that's not yours to fuse, note that the server verify if it contains the drop policyID, if that happens it would be necessary to manually remove the nft with the different policyID from the wallet, ill leave a transaction script in utils for that, then you should just input the asset id of the nft that someone dumb sent you 😂

PLAYING WITH IT

All is about your relation with your comunity, now that you know how the fuse works, and how it might go crash, you can set it working or not turning the cron job on/off, remember that you can call fuse route in order to manually fuse the tokens in the fuse wallet...
All is about your relation with your community, make expending in your nft a thing that is fun.

ALWAYS ONLINE

with a cron job running, i think you might like to set it as always online, so people can fuse til they get what they want (note that they pay for a new token every time they fuse one), i'll let the fuse always online by the time you're reading this, in order to change or turn it off just text me.

CRONJOB EVENT OR MANUALLY

We can play a little and make it like an event, like... telling the comunity the fuse addresss and saying that every day at midnight the server will fuse every bub that is the fuse wallet and send then a high rarity from who sent.... just need to set up the cronjobs or do it manually, calling the route twice should fuse every token in the wallet :3 (in case it didn't call the route till it do but remember the cardano-cli lag)

ADD NEW PROJECTS

```
unction getRandomIndexByProbability
    let r = Math.random(),
        index = probabilities.length

    probabilities.some( function ( p
        if ( r < probability ) {
            index = i;
            return true;
        }
        r -= probability;
    } );
    return index;

 robabilities = [ 0.5, 0.3, 0.15
```

EASY, BUT NOT QUITE

in the dashboard nothing is required

i've made the dashboard in a way that it generates one card per collection.... Therefore coming with a new collection requires some attention, new policyID? metadata object, cronjob and route handler... Well after intense reading of the guide as well the code, you should be able to come with a new collection yourself (or by someone you hired, even me xD)

FIRST COLLECTION

i've made by hand

note in the server that it has a folder called metadata, and there is a file metadata.js, it contains the metadata of each one of the 5555 bubz in the collection, it exists only there, and nothing besides the autoMint.js in mint/controller uses it, coming with a new collection require creating a new metadata obj, filtering by metadata requires caching the metadata locally in the dashboard, the database just stores the index, to prevent storing massive amount of data unnecessary

THE WALLETS TO SEND

DROP WALLET

addr1qypcg9fhxwwvpnfursxv8245qkx8fkgfeqnmjeaavfcmfn
7r65a2wte3jxjlf08z7ks4h9pdvvwnp0fcs3h6nlqw742sv2xxgy

Fuse wallet

addr1qx8fw9u6uged08dk6g7qqz9vns6ncw0wvmauxyjxd
0k3rnpmpfdfxa807m0k3yth9hcldlz56gwmzdjdktl9kuf9uu
0s0l7h4f

CONTACT INFORMATION

Phone Number

55 48 9 9150 7211

Email Address

luis.gabrielrovaris@gmail.com

github / discord

LRovaris #4065