Content

FiveM/ GTA Screenshot: MYG Coin Icon and the held amount of ingame-Database-MYG Coins

Possible MYG Coin Integration / GTA FiveM Server

You will need:

- your own FiveM Server (local installation or remote)
  e.g. files from: https://runtime.fivem.net/artifacts/fivem/build_server_windows/master/

  MYG Coin Plugin works with the standard template (ESX / txAdmin).
  Latest successful test with: Build 5733

- the MYG Coin Plugin from mygcoin.com for FiveM (or visit our Discord)

- MySQL / php MyAdmin Database and for FiveM the „mysql-async“ Plugin
  https://github.com/brouznouf/fivem-mysql-async

You will need to know how to setup a FiveM Server, how to handle a Database like MySQL via phpMyAdmin and (optional) to code in php or any other language. This is NOT part of this paper or Discord-Discussions!

Base idea for implementation

In this scenario we just want to show the MYG Coin Icon inside the game FiveM and print the Database value for the current **ingame**-MYG Coins owned by the user.

IMPORTANT: **ingame-MYG Coins are only values in the database. These are not the „real“ MYG Coins inside a real MYG crypto wallet!**

For the transfer of ingame-MYG coins to a real MYG wallet you will need to subtract the amount of ingame-MYG Coins from the current value in your Database (write it back) and trigger a https-Request against the Webwallet with 2FA access. This means there have to be enough MYG Coins in your Webwallet to handle the transfer and pay the user.

For details on 2FA and the Webwallet watch the Youtube-Videos:

https://www.youtube.com/watch?v=GdeUd1mdX9E
https://www.youtube.com/watch?v=iwF0inmDT9o
https://www.youtube.com/watch?v=-d_LsZoNgOw

Breaking down the URL-Parameters:

Base-URL (not to be changed): https://wallet.mygcoin.com/streamer-giveaway.php?

from=HGFHJHGBJHBGJH -> Enter your 2FA-Key here (**do not show this key public**!!!)

to=MYGADRXYZ12345 -> the receiving MYG-Coin address from the user to send the Coins to

payout=4 -> the amount you want to pay to the user/ or random value

Case:
- *Number* (e.g. 4) = pays exactly this amount of coins
- random = pays a random amount of MYG Coins from your wallet between 10 and 100 MYG Coins.
- If no value (means it is kept empty) is provided, it will pay the standard of 1 MYG Coin

Another way, which avoids using the Webwallet:

You can use the MYG-Cli.exe (for Windows) or via MYG-CLI for Linux.
In your mygcoin.conf you will have to disable staking and enable accounts (for this wallet with labels):

```
enableaccounts=1
staking=0
```

Be sure to know how the concept of accounts works. It is deprecated and can cause problems used the wrong way!

Example Call:

```
mygcoin-cli.exe sendtoaddress MbWGW7RuKkQNtVzaM1BzqG2YdNbirXU2wF 2.0
```

Returns the transaction-ID like:

694839431c47334a97c69096b9627d3702e2710e0f76b8658845ede28aec8dbe

Which you could lookup in the MYG-Explorer:

https://mygexplorer.com/tx/694839431c47334a97c69096b9627d3702e2710e0f76b8658845ede28aec8dbe

Or:

https://chainz.cryptoid.info/myg/tx.dws?694839431c47334a97c69096b9627d3702e2710e0f76b8658845ede28aec8dbe.htm

In php it could look like this:

```
$output=null;

$retval=null;

exec('C:\Users\MYGUser\Desktop\mygcoin-qt-windows-v2>

mygcoin-cli.exe sendtoaddress MbWGW7RuKkQNtVzaM1BzqG2YdNbirXU2wF 2.0', $output, $retval);

echo "Returned with tx-id $retval and output:\n";

print_r($output);
```

See basic php-examples on the MYG Website.

Remember to always check against possible security risks, or against malicious input, or other errors that might occure! All examples in this document and other official MYG papers like the Integration-Paper are only very basic examples or PoC, that need to be secured and double checked and enhanced by you (the developer)! A „Move“ via CLI for example, used to move coins from label/account A to B, needs the dot for the amount (e.g. 15.0 – NOT 15 !!!) and the available balance should always be checked before any transfer! Otherwise sending the amount of „15“ instead „15.0“ could become „150“ MYG coins send! Take care that a balance is never allowed to become negative – so, always double check your codes/ implementation!

Here are some further examples for CLI calls:

Show MYG Address for a „label“:
mygcoin-cli.exe getaddressesbyaccount "Max Muster"

Returns an array of MYG-Addresses belonging to this „label“.

Create MYG Address for a „label“:
mygcoin-cli.exe getnewaddress "Max Muster"

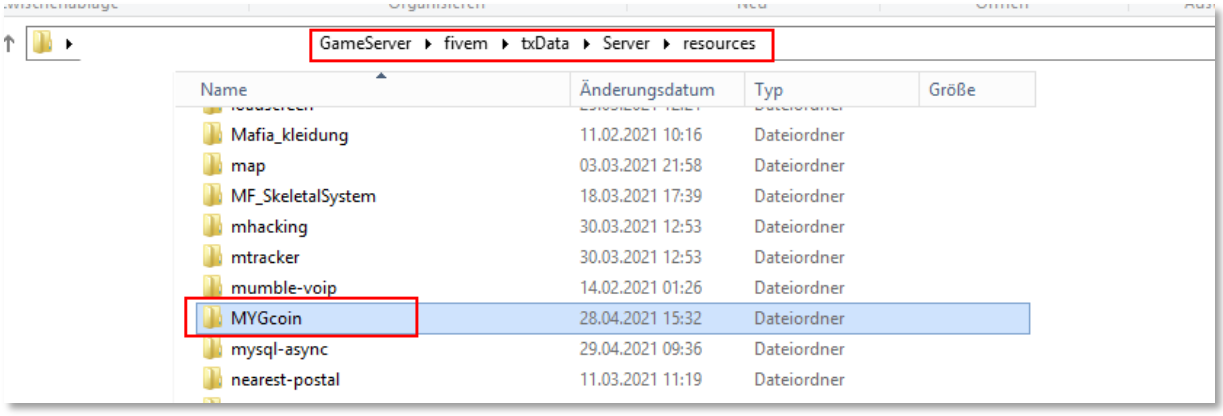
Returns a newly created MYG-Addresses for this „label“.

Check Balance for a „label“:
mygcoin-cli.exe getbalance "Max Muster"

Returns the amount of MYG Coins held by this „label“.

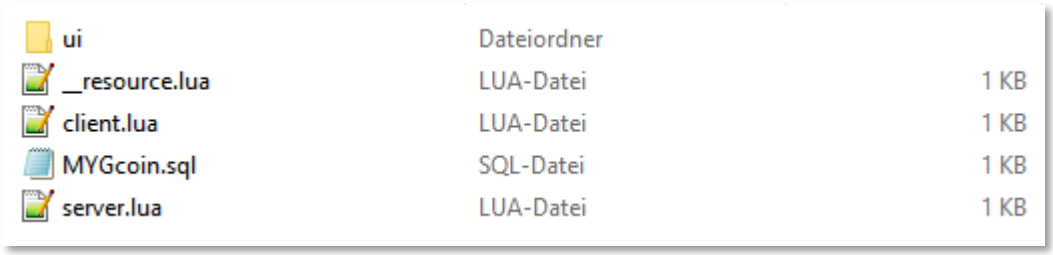
Folders and Config of the MYG FiveM-Plugin

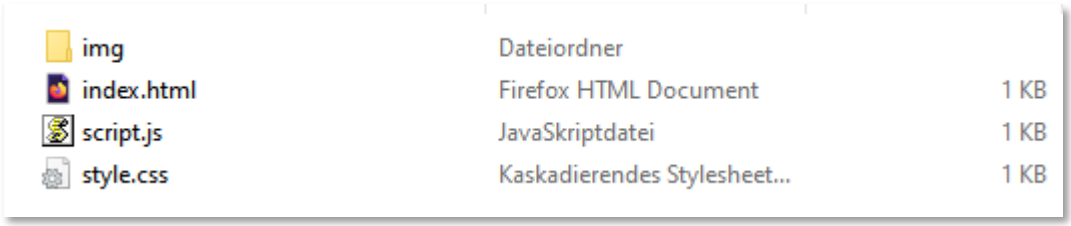
Place the MYGcoin Plugin into your FiveM Folder:

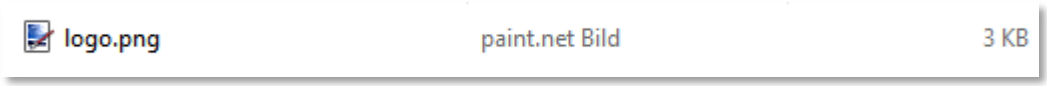


Make sure to start the Plugin in your Server Config File:

Inside the MYGCoin-FiveM-Plugin.zip are these files:







The MYGcoin.sql is needed to insert the MYG Database structure and base-values into your existing ESX Database/Table (Table: users). Please import this file via phpMyAdmin after ESX is installed.

How to make an ingame usecase-connection?

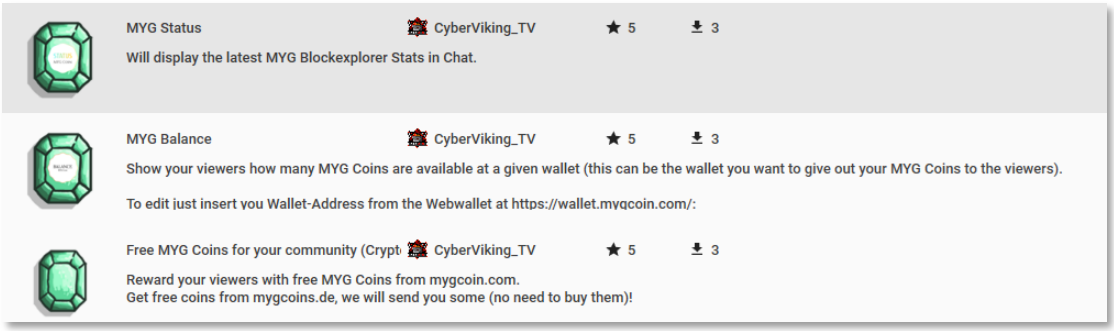
Once the Plugin is installed and running, it is all up to you, to design the internal usecase around the „fake“ – or let us say „only database MYG Coin“ – so it can be deposited/ withdrawn or ingame converted.

You might want to have the players earn ingame-MYG Coins via jobs in FiveM, or simply let them farm the MYG Coins at special interaction points, from your website, etc. etc.

At this point there is no interaction with the real MYG Wallet. In case you want to enable payouts to the users real MYG Wallet (on his PC or Webwallet) make sure you own enough real MYG Coins to make the payout possible from your Webwallet! If you own 50.000 MYG in your Webwallet, you should not have a higher circulating supply in your game! Imagine more than your 50.000 MYG are held by players and they want to pay them out at once. Always back your ingame MYG with REAL MYG Coins!

You run a Stream on Twitch? Use our MixItUp-Bot Community commands and your Viewers can farm REAL MYG right there and transfer it back into your game (check out our php examples for this).

Screenshot MixItUp-MYG Community Commands:

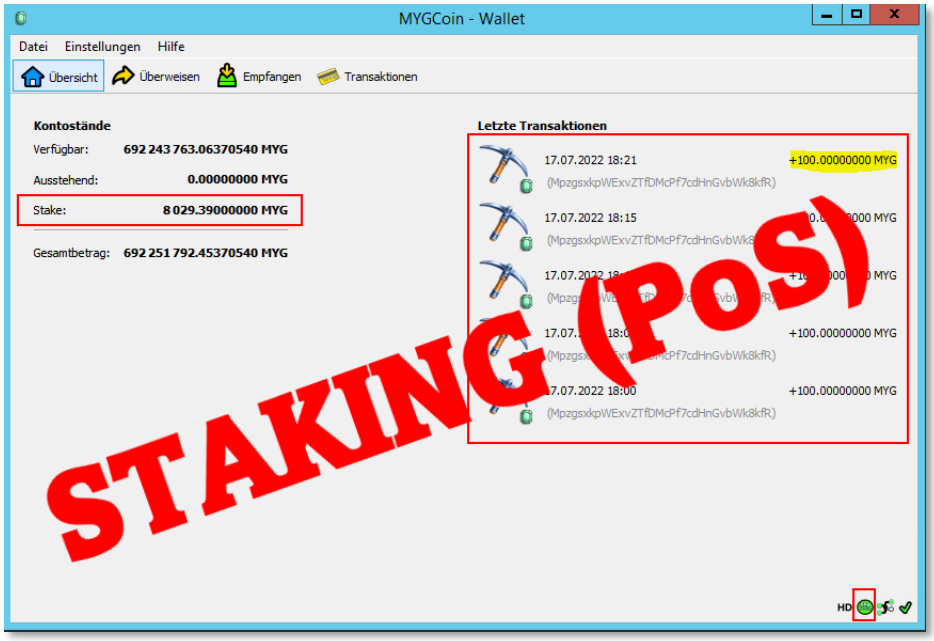


The bigger picture / MYG Coin – a gaming game changer

Once you understand how it is possible to integrate MYG Coins into your game and add value to it, you will notice that having the option to deposit and withdraw the MYG Coins is a real gamechanger!

As a Player you could play game A and earn MYG there, withdraw them to your MYG-Wallet on your PC/Webwallet and deposit those MYG coins in game B, to use them there. This way you can bridge cross plattform, cross game and cross gameserver and always use this MYG Coins where you want and need them!

PoS (Proof of Stake) makes it possible for you to stake the amount of MYG Coins in your PC/Linux Wallet (not possible in the Webwallet!) and accumulate more coins:



A List of MYG CLI Help / Commands

mygcoin-cli.exe help

== Blockchain ==

getbestblockhash

getblock "hash" (verbose)

getblockchaininfo

getblockcount

getblockhash index

getblockheader "hash" (verbose)

getchaintips

getdifficulty

getmempoolancestors txid (verbose)

getmempooldescendants txid (verbose)

getmempoolentry txid

getmempoolinfo

getrawmempool (verbose)

gettxout "txid" n (includemempool)

gettxoutproof ["txid",...] (blockhash)

gettxoutsetinfo

verifychain (checklevel numblocks)

verifytxoutproof "proof"

== Control ==

getinfo

help ("command")

stop

uptime

== Generating ==

generate numblocks (maxtries)

generatetoaddress numblocks address (maxtries)

== Mining ==

checkkernel [{"txid":txid,"vout":n},...] [createblocktemplate=false]

getblocktemplate (TemplateRequest)

getmininginfo

getnetworkhashps (blocks height)

getstakinginfo

prioritisetransaction <txid> <priority delta> <fee delta>

submitblock "hexdata" ("jsonparametersobject")

== Network ==

addnode "node" "add|remove|onetry"

clearbanned

disconnectnode "node"

getaddednodeinfo dummy ("node")

getconnectioncount

getnettotals

getnetworkinfo

getpeerinfo

listbanned

ping

setban "ip(/netmask)" "add|remove" (bantime) (absolute)

== Rawtransactions ==

createrawtransaction [{"txid":"id","vout":n},...] {"address":amount,"data":"hex"

,...} (locktime)

decoderawtransaction "hexstring"

decodescript "hex"

fundrawtransaction "hexstring" (options)

getnormalizedtxid "hexstring"

getrawtransaction "txid" (verbose)

sendrawtransaction "hexstring" (allowhighfees)

signrawtransaction "hexstring" ([{"txid":"id","vout":n,"scriptPubKey":"hex","re

deemScript":"hex"},...] ["privatekey1",...] sighashtype)

== Util ==

createmultisig nrequired ["key",...]

estimatefee nblocks

estimatepriority nblocks

estimatesmartfee nblocks

estimatesmartpriority nblocks

signmessagewithprivkey "privkey" "message"

validateaddress "bitcoinaddress"

verifymessage "mygcoinaddress" "signature" "message"

== Wallet ==

abandontransaction "txid"

abortrescan

addmultisigaddress nrequired ["key",...] ("account")

backupwallet "destination"

burn <amount> [hex string]

dumpprivkey "mygcoinaddress"

dumpwallet "filename"

encryptwallet "passphrase"

getaccount "mygcoinaddress"

getaccountaddress "account"

getaddressesbyaccount "account"

getbalance ("account" minconf includeWatchonly)

getnewaddress ("account")

getrawchangeaddress

getreceivedbyaccount "account" (minconf)

getreceivedbyaddress "mygcoinaddress" (minconf)

gettransaction "txid" (includeWatchonly)

getunconfirmedbalance

getwalletinfo

importaddress "address" ("label" rescan p2sh)

importprivkey "bitcoinprivkey" ("label" rescan)

importprunedfunds

importpubkey "pubkey" ("label" rescan)

importwallet "filename"

keypoolrefill (newsize)

listaccounts (minconf includeWatchonly)

listaddressgroupings

listlockunspent

listreceivedbyaccount (minconf includeempty includeWatchonly)

listreceivedbyaddress (minconf includeempty includeWatchonly)

listsinceblock ("blockhash" target-confirmations includeWatchonly)

listtransactions ("account" count from includeWatchonly)

listunspent (minconf maxconf ["address",...])

lockunspent unlock ([{"txid":"txid","vout":n},...])

move "fromaccount" "toaccount" amount (minconf "comment")

removeprunedfunds "txid"

reservebalance [<reserve> [amount]]

sendfrom "fromaccount" "tomygcoinaddress" amount (minconf "comment" "comment-to

")

sendmany "fromaccount" {"address":amount,...} (minconf "comment" ["address",...

])

sendtoaddress "mygcoinaddress" amount ("comment" "comment-to" subtractfeefromam

ount)

setaccount "mygcoinaddress" "account"

settxfee amount

signmessage "mygcoinaddress" "message"

See also for educational purposes (might contain different commands than used by MYG):

https://chainquery.com/bitcoin-cli#wallet

https://github.com/BlockchainCommons/Learning-Bitcoin-from-the-Command-Line/blob/master/04_1_Sending_Coins_The_Easy_Way.md