Potato Sniper Bot

Contact Info

Telegram Chat: https://t.me/PotatoSniperBot
Dev-Telegram: @poormanmentality
Dev-Discord: Anonymous101#5251

GitHub: https://github.com/PotatoSniper/Potato_Bot

Disclaimer

Mr. Potato holds no responsible or reliable for any error or losses, if you persist on using this tool then you agree to hold full responsibility and liability for all consequences. If you are uncomfortable or disagree with any of the terms of this policy, please discontinue use of Potato Sniper Bot.

Warning

⚠ Only use test wallet and never use main wallet ⚠

Table Of Contents

Shitty Telegram Sniper Bot

⚠ Compatible with public node but make sure you set delay high enough so it doesn't hit rate limit ⚠

All purchases will utilize governance token (*BNB/ETH/CRO/MATIC/FTM*)

Supported Chains	Functionality
Binance Smart-Chain ➢ PancakeSwap ➢ ApeSwap ➢ BabySwap ➢ KnightSwap ➢ MDEXSwap **Ethereum** ➢ UniSwapV2 ➢ SushiSwap ➢ ShibaSwap **Chronos** ➢ MeerkatSwap **Polygon** ➢ QuickSwap ➢ PolycatSwap **Fantom** ➢ SpookySwap ➢ SpiritSwap **Avalanche** ➢ TraderJoe ➢ Pangolin **Milkomeda** ➢ MilkySwap ➢ OsccamxSwap ➢ MuesliSwap **Metis** ➢ TethySwap	**Honeypot Smart Contract** Used to measure real time buy and sell tax and trading status of target contract. **Call Channel Sniper Mode** ✓ Listens to 1 or multiple channels at once and trigger buy whenever a contract address is detected and passes all token checks. Once buy is successful, that address will be blacklisted and never buy again if it appears afterward. Auto sell at x profit and x percent stop loss. **Fair Launch Sniper (Groups) Mode** ✓ Scrape chat data and reconstruct data to contract address ✓ Convert LP pair address to contract address ✓ Download txt file > read content > construct contract address ✓ Filter greeting messages ✓ Auto sell at x profit **Tradable Mode** ✓ Buy when target contract is tradable/sellable and clears all checking from HoneyPot Contract. ✓ Auto sell at x profit **Limit-Trade Mode** ✓ Buy/Sell at predefined price/multiple price target and amount.

OS Download / Run Guide

Linux

⚠ Replace ***filename*** with actual name of the file ⚠

⚠ Replace ***NameOfDirectory*** with actual name of the folder ⚠

Download & Edit permission

1. cd /home/
2. git clone https://github.com/PotatoSniper/Potato_Bot
3. cd Potato_Bot
4. apt-get install unzip
5. unzip ***filename***
6. cd ***NameOfDirectory*/*/**
7. chmod u+x ***filename***

Modify config.json file

1. nano config.json To save: [**CTRL+X** > **Y** > **ENTER**]

Run

1. ./***filename***

Clean-Up (Optional)

1. mv ***filename*** /home/***NameOfDirectory***/
2. cd /home/
3. rm -r Potato_Bot/

Window

1. Download files at https://github.com/PotatoSniper/Potato_Bot
2. Modify config.json file
3. Run

Mac

1. Not sure, try Linux setup and change **Linux** to **Mac**

Call Channel Sniper Mode

⚠ Do Not Use Wallet While Bot Is Running ⚠

Initial Setup

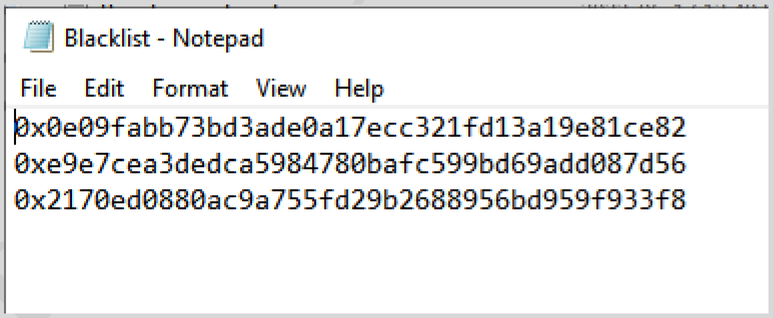
1. Refer to **Config.json setup section** in this manual to fill in missing information.
2. Run **Deployer** and deploy management contract, copy and paste return address under "**management_contract"** in **config.json** file.

Bot Setup

3. Run **Shitty Telegram Sniper**.
4. Enter **1** to start telegram login authentication process. Local session will be saved for future logins, no longer required to enter password and verification code.
5. Enter **2** to print all chat ID, copy and paste those target channel ID under **"Channel_IDs_To_Monitor"** in **config.json** file.
6. (**Optiona**l), Enter **3** to print out all messages from those channels just to make sure it's capturing and running properly. (This mode does not buy, simply printing out all messages.)
7. Enter **4** to start Call Channel Sniper.

Blacklist.txt

You can manually blacklist token address; bot will ignore it if these tokens appear in any of the target call channels. Below is an example input format, make sure there are no spaces before or after the contract address.



```
0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82
0xe9e7cea3dedca5984780bafc599bd69add087d56
0x2170ed0880ac9a755fd29b2688956bd959f933f8
```

Features

- ✓ Honey Pot Check.
- ✓ Tax Check
- ✓ Minimum LP Check
- ✓ Maximum LP Check
- ✓ Auto Sell at X Profit
- ✓ Auto Sell at Stop Loss %

Fair Launch Sniper Mode

⚠ Do Not Use Wallet While Bot Is Running ⚠

Initial Setup

1. Refer to **Config.json setup section** in this manual to fill in missing information.
2. Run **Deployer** and deploy management contract, copy and paste return address under "**management_contract"** in **config.json** file.

Bot Setup

3. Run **Shitty Telegram Sniper**.
4. Enter **1** to start telegram login authentication process. Local session will be saved for future logins, no longer required to enter password and verification code.
5. Enter **2** to print all chat ID and load all chat id to local session.
6. Enter **5** to start Fair Launch Sniper configuration process.
7. Copy and Paste group ID to terminal and press **ENTER**.

This mode works for both public and private groups. Only recognize messages send or edited messages from admins, all non-admin and bot messages will be ignored.

Features

✓ Scrape chat data and reconstruct data to contract address
✓ Convert LP pair address to contract address
✓ Download txt file > read content > construct contract address
✓ Honey Pot Check.
✓ Auto Sell at X Profit

Limit Trade Mode

⚠ Do Not Use Wallet While Bot Is Running ⚠

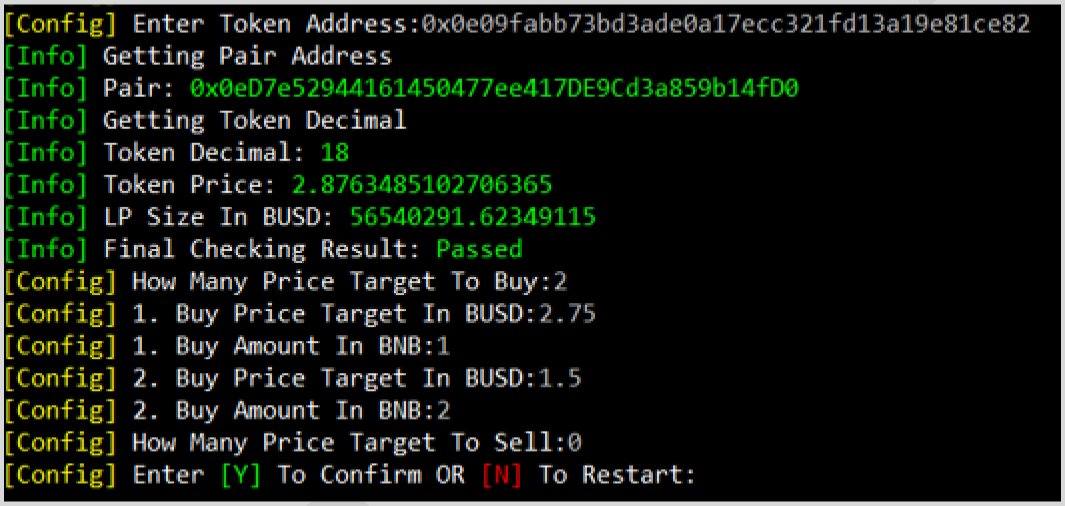
Initial Setup

1. Refer to **Config.json setup section** in this manual to fill in missing information.
2. Run **Deployer** and deploy management contract, copy and paste return address under "**management_contract"** in **config.json** file.

Bot Setup

3. Run **Shitty Telegram Sniper**.
4. Enter **7** to start configuration process.

Buy Example.



```
[Config] Enter Token Address:0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82
[Info] Getting Pair Address
[Info] Pair: 0x0eD7e52944161450477ee417DE9Cd3a859b14fD0
[Info] Getting Token Decimal
[Info] Token Decimal: 18
[Info] Token Price: 2.8763485102706365
[Info] LP Size In BUSD: 56540291.62349115
[Info] Final Checking Result: Passed
[Config] How Many Price Target To Buy:2
[Config] 1. Buy Price Target In BUSD:2.75
[Config] 1. Buy Amount In BNB:1
[Config] 2. Buy Price Target In BUSD:1.5
[Config] 2. Buy Amount In BNB:2
[Config] How Many Price Target To Sell:0
[Config] Enter [Y] To Confirm OR [N] To Restart:
```

This example illustrates buying CAKE token at 2 different price points, if price falls to $2.75 then buy 1 BNB worth of CAKE, if price falls to $1.5 buy 2 BNB worth of CAKE. Sell is disabled in this scenario with 0.

Sell Example.

- ***Enter a sell price target, bot will sell all balance if price is greater than or equal to defined target. 0 to disable.***
- ***Enter a stop loss price target. If price falls below or equal to defined target, bot will sell all balance and terminate all buy/sell actions. 0 to disable.***

If both buy and sell actions were completed successfully, bot will restart the buy and sell process. This mode is used for swing trading.

Tradeable Mode

⚠ Do Not Use Wallet While Bot Is Running ⚠

Initial Setup

1. Refer to **Config.json setup section** in this manual to fill in missing information.
2. Run **Deployer** and deploy management contract, copy and paste return address under "**management_contract"** in **config.json** file.

Bot Setup

3. Run **Shitty Telegram Sniper**.
4. Enter **6** to start configuration process.
5. Paste token address in terminal and press **ENTER**.

Features

- ✓ Honey Pot Check.
- ✓ Tax Check
- ✓ Auto sell at x profit

This mode continuously monitors target contract status and execute buy transactions if passes all checking. Buy transaction will only initiate if target contract is available for buy and sell and tax is lower than or equal to config.json file configuration.

Config.json File Set-Up

Telegram Data Config

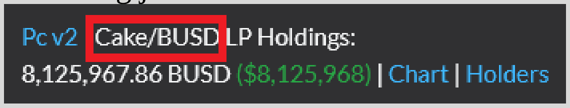
"apiID" ***"apiHash"***	Method 1 (PC). 1. Obtain from https://my.telegram.org/auth Method 2 (Android). 1. Download Opera Mini from Play Store 2. In Opera Mini Browser enter https://my.telegram.org/auth
"Channel_IDs_To_Monitor"	Enter channel id here to monitor. Used for Call Channel Sniper Mode Only
"track_edited_message"	Turn on monitor edit messages. Only accepts **true** or **false** as input.

Block Chain Data Config

"block_chain"	Defines the chain symbol for bot recognition. Refer to Address.txt on GitHub for supported input.
"url"	Enter Node endpoint url.
"router"	Defines DEX router address to use. Refer to Address.txt on GitHub for supported input.
"factory"	Defines DEX factory address to use. Refer to Address.txt on GitHub for supported input.
"buy_amount_in_bnb"	Defines the buy amount in governance token.
"gas"	Defines the gwei value used in every transaction.
"sell_gas"	Defines the gwei value used in every sell transaction.
"gaslimit"	Defines the maximum allowed gas one transaction can use.
"max_allowed_buy_tax" ***"max_allowed_sell_tax"***	Checks token tax, if tax is equal to or below defined value then execute buy transaction.
"governance_token_addr"	Defines governance token address to use. Refer to Address.txt on GitHub for supported input.

"intermediate_token"	Defines intermediate token address to use. See example below. **false** to disable.

Example.
Cake token paired with BUSD. Enter BUSD address under ***"intermediate_token"*** parameter in config.json file. CASE SENSITIVE.


Pc v2 Cake/BUSD LP Holdings:
8,125,967.86 BUSD ($8,125,968) | Chart | Holders

Block Chain	Governance Token
Binance Smart-Chain	BNB
Ethereum	ETH
Chronos	CRO
Polygon	MATIC
Fantom	FTM
Avalanche	AVAX

Call Channel Sniper Config

"min_lp_size_in_bnb"	Defines minimum size of Liquidity Pool.
"max_lp_size_in_bnb"	Defines maximum size of Liquidity Pool.
"auto_sell_all_balance_x_profit"	Defines the current balance worth in comparison to initial purchase amount. **false** to disable. If you bought for 1 BNB and want to auto sell when balance is worth 2 BNB, then put **2** here.
"stop_loss_percent"	Defines x % down from initial purchase before sell full balance. ***Warning. If token buy tax is 20% and you have this as 20% then it would sell instantly because your current balance is worth 20% less than buy amount***
"max_sell_attempt"	Defines maximum allowed fail sell attempts for each token address
"price_check_frequency_in_milliseconds"	Defines price update frequency
"get_balance_delay_in_milliseconds"	Defines retry delay to get balance if failed.

Group Sniper Config

"auto_sell_all_balance_x_profit"	Defines the current balance worth in comparison to initial purchase amount. **false** to disable. If you bought for 1 BNB and want to auto sell when balance is worth 2 BNB, then put **2** here. This configuration also applies for **Tradeable Mode**.

"enable_download_txt_file_and_read"	Accepts **true** or **false**. If enable it will download .txt file and read for data.
"price_check_frequency_in_milliseconds"	Defines price update frequency

Limit Trade Config

"min_lp_size_in_usd"	Defines minimum size of Liquidity Pool.
"max_buy_attempt"	Defines maximum allowed fail attempts for buy transaction
"max_sell_attempt"	Defines maximum allowed fail attempts for sell transaction
"price_check_frequency_in_milliseconds"	Defines price update frequency

Shitty Mempool Sniper Bot

⚠ Not Compatible with public node. Contact dev to setup a private endpoint ⚠

All purchases will utilize governance token (*BNB/ETH/CRO/MATIC/FTM*)

Supported Chains	Functionality
Binance Smart-Chain ➢ PancakeSwap ➢ ApeSwap ➢ BabySwap ➢ KnightSwap ➢ MDEXSwap **Ethereum** ➢ UniSwapV2 ➢ SushiSwap ➢ ShibaSwap **Polygon** ➢ QuickSwap ➢ PolycatSwap **Fantom** ➢ SpookySwap ➢ SpiritSwap **Milkomeda** ➢ MilkySwap ➢ OsccamxSwap ➢ MuesliSwap **Metis** ➢ TethySwap	**Same Block Snipe** ✓ NON-BNB, BNB, DxSale, PinkSale, GemPad, Buy On Function Name. **Blacklist and High Tax Prevention** ✓ Using test buy condition to measure blacklist and tax status in the same transaction and revert it true. **By Pass Cool Down and Max Amount** ✓ utilizes smart contract as wallets to make seperate buy transactions to by pass buy cooldown and max transaction amount. (All buy from all wallets will be done within one transaction). **Anti-Rug Mechanism** ✓ Monitors Mempool for all remove LP transactions related to target token, if detected, the bot will attempt to front run it with 2x gas and sell before it. ✓ Monitor Mempool based on specified function name. If specified function is being called to the target contract, the bot will attempt to front run with 2x gas and sell before it. **Tradeable Mode** ✓ This mode checks the target contract status and ensure it is buy and sell able and all tax parameter is met before executing a buy transaction. Slow but safe. Best result is current block + 2 away from when status is clear.

Set-Up

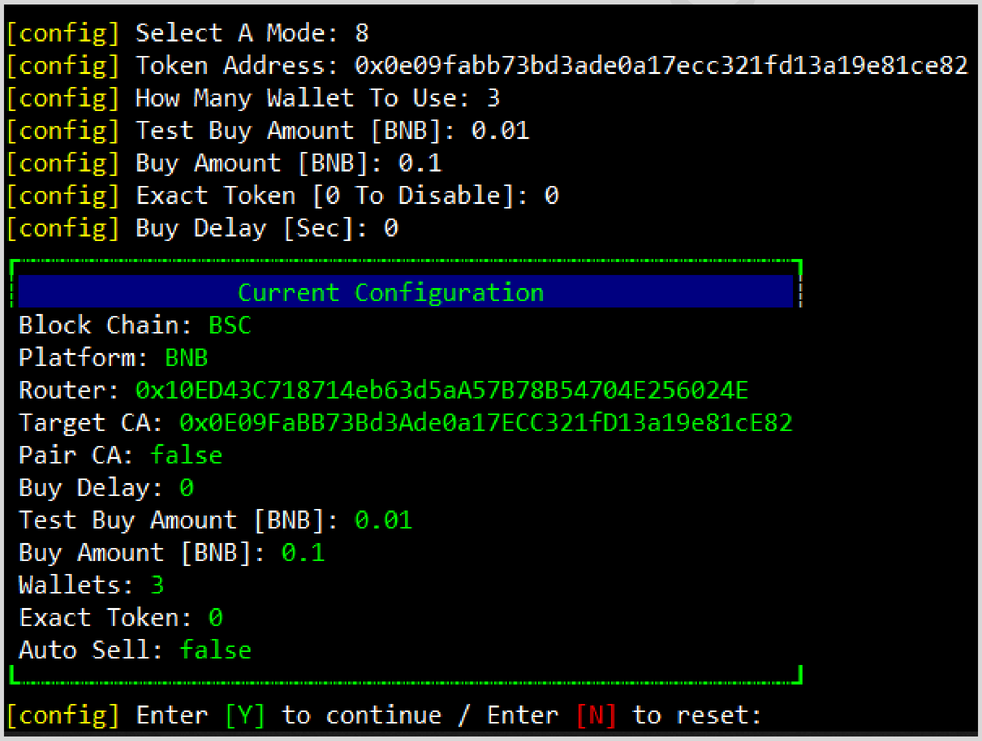
Initial Setup

1. Refer to **Config.json setup section** in this manual to fill in missing information.
1. Run **Deployer** and deploy management contract. Enter **1** to deploy management contract. **(Cost around 4.2 million gas, make sure your gas limit is at least 5 Million).** Copy and paste return address under "**management_contract"** in **config.json** file.
2. Run Deployer and create multi-wallet. Enter **2** to create multi-wallets, each wallet is an individual smart contracts used for buying and selling. **(Each wallet takes around 1.1 million gas to deploy. Make sure your gas limit is enough for the number of wallets you are deploying)**

Highly suggest using the lowest gas value to deploy contract to save money. BSC use 5 gwei.

Only need to deploy contract once. ***Do Not*** *deploy contract every time using the bot.*

Buy Transaction Configuration



```
[config] Select A Mode: 8
[config] Token Address: 0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82
[config] How Many Wallet To Use: 3
[config] Test Buy Amount [BNB]: 0.01
[config] Buy Amount [BNB]: 0.1
[config] Exact Token [0 To Disable]: 0
[config] Buy Delay [Sec]: 0

                Current Configuration
 Block Chain: BSC
 Platform: BNB
 Router: 0x10ED43C718714eb63d5aA57B78B54704E256024E
 Target CA: 0x0E09FaBB73Bd3Ade0a17ECC321fD13a19e81cE82
 Pair CA: false
 Buy Delay: 0
 Test Buy Amount [BNB]: 0.01
 Buy Amount [BNB]: 0.1
 Wallets: 3
 Exact Token: 0
 Auto Sell: false

[config] Enter [Y] to continue / Enter [N] to reset:
```

1. Select a mode of operation
2. Enter token address
3. Enter number of wallets to use. Do not enter more than you deployed.
4. Enter test buy amount. This will test for blacklist and check for buy + sell tax condition to ensure you don't get rekt. **0** to disable.
5. Enter total amount of governance token to buy for all wallets. This value will be spitted with x wallets used. For example, 5 wallets and buy amount 1 BNB, each wallet will buy 0.2bnb

worth of tokens with buy exact eth mode. To enable exact eth mode, enter 0 for buy exact token.

6. Exact token mode. Enter a value for exact token amount if you wish to get an exact amount of token with a given buy amount value. For example, 5 wallets buy amount 1 BNB and want exactly 1000 tokens in each wallet. Each wallet will attempt to buy 1000 tokens with 0.2 BNB, if one of the wallets failed to buy due to underpricing then the entire transaction will fail and get reverted, make sure you give more BNB just in case of underprice error, all remaining/extra BNB will be transfer back to original wallet at the end of the transaction. **(Only Available in Paid Version).**
7. Enter a buy delay value for projects with dead blocks. **0** to disable.

NON-BNB Mode

- This mode is used to monitor all add LP status of all non-governance pairs added by defined **router** defined in **config.json** file. **Only works if trading is enabled at the time of adding LP.**

BNB Mode

- This mode is used to monitor all add LP status of all governance pairs added by defined **router** defined in **config.json** file. **Only works if trading is enabled at the time of adding LP.**

DXSale Mode

- This mode is used to monitor DxSale add LP status for target token address. **Only works if trading is enabled at the time of adding LP.**

PinkSale Mode

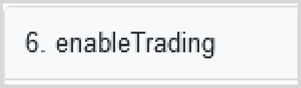
- This mode is used to monitor PinkSale add LP status for target token address. **Only works if trading is enabled at the time of adding LP.**

GemPad Mode

- This mode is used to monitor GemPad add LP status for target token address. **Only works if trading is enabled at the time of adding LP.**

Buy On Function Name Mode

- Enter the function name you wish to execute buy transaction when the function is being called on the token contract. For example, enableTrading. Case sensitive. **This Mode is used when trading is disabled at the time of adding LP.**



6. enableTrading

Buy Token Mode

- Regular buy.

Sell Token Mode

- Regular sell.

Tradeable Mode

- This mode continuously monitors target contract status and execute buy transactions if passes all checking. Buy transaction will only initiate if target contract is available for buy and sell and tax is lower than or equal to **config.json** file configuration.

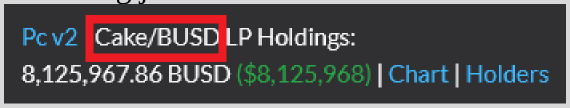
Config.json File Set-Up

Block Chain Data Config

"block_chain"	Defines the chain symbol for bot recognition. Refer to Address.txt on GitHub for supported input.
"url"	Enter Node endpoint url.
"router"	Defines DEX router address to use. Refer to Address.txt on GitHub for supported input.
"factory"	Defines DEX factory address to use. Refer to Address.txt on GitHub for supported input.
"buy_gas"	Defines the gwei value used in every transaction.
"sell_gas"	Defines the gwei value used in every sell transaction.
"gaslimit"	Defines the maximum allowed gas one transaction can use.
"auto_sell_at_x_profit"	Defines the current balance worth in comparison to initial purchase amount. **false** to disable. If you bought for 1 BNB and want to auto sell when balance is worth 2 BNB, then put **2** here.
"auto_sell_balance_percent"	Defines percent of balance to sell when **"auto_sell_x_profit"** is met. **Only available in PAID version.**

auto_sell_max_attempts	Defines maximum allowed fail sell attempts.
"buy_tax" ***"sell_tax"***	Checks token tax, if tax is equal to or below defined value. Only applies for Tradeable Mode.
"governance_token_addr"	Defines governance token address to use. Refer to Address.txt on GitHub for supported input.
"intermediate_token"	Defines intermediate token address to use. See example below. **false** to disable.

Example.
Cake token paired with BUSD. Enter BUSD address under ***"intermediate_token"*** parameter in config.json file. CASE SENSITIVE.

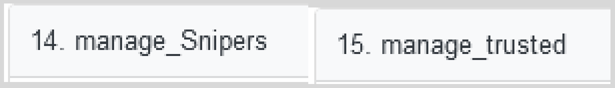

Pc v2 Cake/BUSD LP Holdings:
8,125,967.86 BUSD ($8,125,968) | Chart | Holders

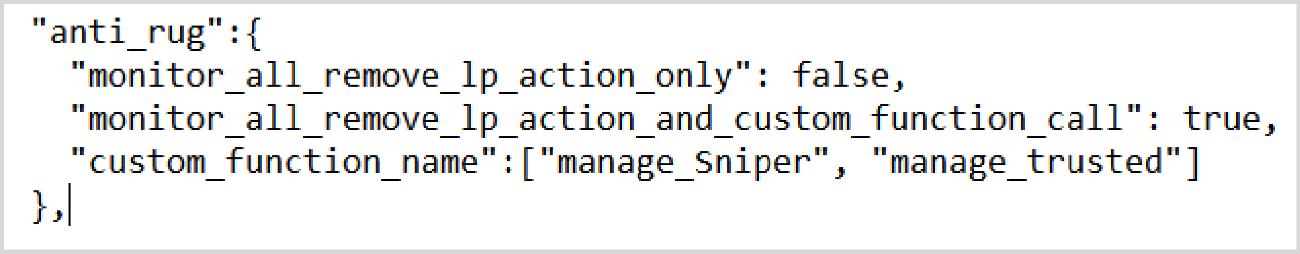
Block Chain	Governance Token
Binance Smart-Chain	BNB
Ethereum	ETH
Chronos	CRO
Polygon	MATIC
Fantom	FTM
Avalanche	AVAX

Anti Rug Config

"monitor_all_remove_lp_action_only"	Monitor mempool for all remove LP transactions related to target token, if detected the bot will attempt to front run the target transaction with 2x gas. Accepts **true** or **false** input.
" monitor_all_remove_lp_action_ and_custom_function_call"	Monitor mempool for all remove LP and specified function call transactions related to target token, if detected the bot will attempt to front run the target transaction with 2x gas. Accepts **true** or **false** input.
"custom_function_name"	Function names to monitor and trigger front run selling. See example below.

Example.


14. manage_Snipers
15. manage_trusted



```
"anti_rug":{
  "monitor_all_remove_lp_action_only": false,
  "monitor_all_remove_lp_action_and_custom_function_call": true,
  "custom_function_name":["manage_Sniper", "manage_trusted"]
},
```

Only turn on one. If both are true, "monitor_all_remove_lp_action_only" will have highest priority.

do_not_touch_unless_required Config

"token_decimal"	Bot will automatically fetch this data, if failed then you are required to enter the decimal value for governance token.
" intermediate_token_decimal"	Bot will automatically fetch this data, if failed then you are required to enter the decimal value for intermediate token paired with target token.
"method_id_of_buy_function_name"	Bot will automatically fetch this data, if failed then you are required to enter the method id of the function name. Expecting 10 digits hexadecimal type.
"method_id_of_custom_function_name"	Bot will automatically fetch this data, if failed then you are required to enter the method id of all function name. Expecting 10 digits hexadecimal type.

Error Log

insufficient funds for gas * price + value

- This error occurs when you don’t have enough to cover gas and buy amount. Below is an example of gas calculation for one transaction if max gas limit is used.

Example.

Config.json file, ***"gas": "5"*** and ***"gaslimit": "2000000".*** Maximum Gas cost would be 0.01 BNB

$$gas\ cost\ in\ BNB = \frac{5}{10^9}\ x\ 2000000 = 0.01\ BNB$$

Nonce is too low

- This error will occur if wallet is used when bot is running. Do not use wallet while bot running.

Fail with error 'PancakeLibrary: INSUFFICIENT_LIQUIDITY'

- This error occurs when you front ran the ADD_LP transaction, this error doesn’t happen frequently but does happen and no fix to it as all bots encounter this error on a fast private node.

Fail with error 'PancakeRouter: EXCESSIVE_INPUT_AMOUNT'

- This error occurs when using exact token mode, it means that your transaction was underprice for the token amount you want to buy. In short, you are paying too little and asked for too much token.

Fail with error 'Pancake: TRANSFER_FAILED'

- This error can occur if your transaction was executed and target contract is not tradeable. Or if you had testbuy and got blacklisted and failed to sell during the blacklist and high tax check process.