

# 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](https://github.com/PotatoSniper/Potato_Bot)

YouTube Demo: <https://www.youtube.com/watch?v=3t74O6JmtNI>

## 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

|  |           |
|--|-----------|
| Potato Sniper Bot Introduction .....                             | 4         |
| <b>Supported Chains</b> .....                                    | <b>4</b>  |
| <b>Functionality</b> .....                                       | <b>4</b>  |
| Setup .....  | 5         |
| <b>Linux</b> .....   | <b>5</b>  |
| <b>Window</b> .....  | <b>5</b>  |
| <b>Mac</b> .....   | <b>5</b>  |
| ContractDeployer .....   | 6         |
| <b>Deploy Contract</b> .....                                     | <b>6</b>  |
| <b>Withdraw</b> .....  | <b>6</b>  |
| PotatoBot .....  | 7         |
| <b>Mempool Mode</b> .....  | <b>7</b>  |
| <b>Telegram Mode</b> .....                                       | <b>8</b>  |
| <i>Chat Link Mode [1]</i> .....                                  | 8         |
| <i>Chat ID Mode [2]</i> .....                                    | 9         |
| <b>Tradable Mode</b> .....                                       | <b>9</b>  |
| <b>Set Fee (Mempool / Telegram/ Tradable)</b> .....              | <b>10</b> |
| <b>Limit Trade Mode</b> .....                                    | <b>10</b> |
| <i>Set Buy Target</i> .....                                      | 11        |
| <i>Set Sell Target</i> .....                                     | 12        |
| Config.json Setup.....   | 13        |
| <b>api ID &amp; Hash</b> .....                                   | <b>13</b> |
| <b>url</b> .....   | <b>13</b> |
| <b>block_chain, router, factory, governance_token_addr</b> ..... | <b>13</b> |
| <b>Intermediate_token</b> .....                                  | <b>13</b> |
| <b>target_owner</b> .....  | <b>13</b> |
| <b>my_contract, hp_contract</b> .....                            | <b>14</b> |
| <b>contract_owner_addr, contract_owner_key</b> .....             | <b>14</b> |
| <b>sell_account_addr, sell_account_key</b> .....                 | <b>14</b> |
| <b>auto_sell config</b> .....                                    | <b>14</b> |
| <b>Telegram_Honeypot, Telegram_Download_File</b> .....           | <b>14</b> |
| <b>target_buytax, target_selltax</b> .....                       | <b>15</b> |

|  |           |
|--|-----------|
| <b>buy_amount, buy_count, test_buy_amount, buy_delay_timer, gas, gaslimit, sell_slippage .....</b> | <b>15</b> |
| <b>limit_trade .....</b>   | <b>15</b> |

Potato Sniper Bot

## Potato Sniper Bot Introduction

⚠ Not compatible with public/limit rated endpoints contact dev to setup a private endpoint ⚠

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

### Supported Chains

#### **Binance Smart-Chain**

- PancakeSwap
- ApeSwap
- BabySwap
- KnightSwap
- MDEXSwap

#### **Ethereum**

- UniSwapV2
- SushiSwap
- ShibaSwap

#### **Chronos**

- MeerkatSwap

#### **Polygon**

- QuickSwap
- PolycatSwap

#### **Fantom**

- SpookySwap
- SpiritSwap

#### **Avalanche**

- TraderJoe
- Pangolin

### Functionality

#### **Buy Smart Contract**

Used for purchase transactions only, checking for real-time blacklist and high tax. It's a anti anti-bot measure. Only for Mempool, Telegram and Tradeable Mode.

#### **Honeypot Smart Contract**

Used to measure real time buy and sell tax and trading status of target contract. Only for Telegram and Tradeable Mode.

#### **Mempool Mode**

- ✓ Scan Mempool

#### **Telegram 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

#### **Tradable Mode**

- ✓ Buy when target contract is tradable/sellable
- ✓ Honeypot checker for buy/sell tax

#### **Limit-Trade Mode**

- ✓ Buy/Sell at predefined price/multiple price target and amount.

*This mode uses "contract\_owner\_addr" for all transactions. Not with Buy Smart Contract.*

## Setup

### Linux

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

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

#### Download & Edit permission

1. cd home/
2. gitclone [https://github.com/PotatoSniper/Potato\\_Bot](https://github.com/PotatoSniper/Potato_Bot)
3. cd Potato\_Bot/
4. unzip Linux.zip
5. cd Linux/
6. chmod u+x ContractDeployer
7. chmod u+x PotatoBot

#### 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](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**

## ContractDeployer

Deployer crashes upon opening > double check > `"url"`, `"contract_owner_addr"` in config.json file.

```
-----
                        Thank You For Using Potato Bot
                        -----
                        [ Contact ]
                        Telegram-Chat: https://t.me/PotatoSniperBot
                        Dev-Telegram: @poormanmentality
                        Dev-Discord: Anonymous101#5251
                        -----
Address: [redacted] Client Status:Free
!!!!!!!!!!!!!! Please Select Mode Of Operation !!!!!!!!!!!!!!!
Deploy Buy-Contract      [1]:
Deploy HoneyPot-Contract [2]:
Withdraw                 [3]:
Exit                     [x]:
```

### Deploy Contract

- `url` parameter in config.json file will determine which chain the contract is deployed.
- All contracts will utilize `contract_owner_addr` and `contract_owner_key` for deployment, make sure you have enough to cover gas fee.  $gas\ fee = \frac{gas}{10^9} \times gaslimit$
- Copy and paste returned contract address to config.json file under `"my_contract"` or `"hp_contract"`

### Example.

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

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

### Withdraw

⚠ Only contract owner (`contract_owner_addr`) can execute this. ⚠

- Withdraw balance from buy contract.
- Make sure contract owner wallet have enough to cover gas fee.
- Restart Bot to load updated config.json file data.

## PotatoBot

PotatoBot crashes upon opening > double check > `"url", "router", "factory", "governance_token_addr", "my_contract", "hp_contract", "contract_owner_addr", "sell_account_addr"` in config.json file.

⚠ Ensure **Contract Owner Wallet** has enough to cover gas fee ⚠

⚠ Ensure **Buy Contract** has enough to cover  $(buy\_amount * buy\_count + test\_buy\_amount)$  ⚠

## Mempool Mode

⚠ **PinkSale/Dxsale** only in BSC ⚠

```
-----
                        Thank You For Using Potato Bot
                        [
Telegram-Chat: https://t.me/PotatoSniperBot
Dev-Telegram: @poormanmentality
Dev-Discord: Anonymous101#5251
                        ]
-----
Address: [REDACTED] Client Status:Free

!!!!!!!!!!!!!! Please Select Mode of Operation !!!!!!!!!!!!!!!

Mempool Snipe [1]:
Telegram Snipe [2]:
Tradable Snipe [3]:
Limit-Trade   [4]:
1

!!!!!!!!!!!!!! Please Select A Launching Platform !!!!!!!!!!!!!!!
Mempool (NON-BNB) [0]:
Mempool (BNB)    [1]:
Monitor Owner    [2]:
PinkSale         [3]:
Dxsale           [4]:
```

- ✓ Scan Mempool for target transaction
- ✓ Auto determine buy gas if `"buy_delay_timer": null`, to ensure buying in block 0.
- ✓ If Add\_LP is detected and added successfully but buy transaction failed, the bot will assume trading is disabled and automatically switch to monitor owner–contract interaction for enable trade action. If `contract_owner` is empty in confirmation step, you must manually add it in config.json file for this to work.
- ✓ Buy from smart contract with governance token.

⚠ Fill in `"intermediate_token"` parameter in config.json file, if target token is pairing with non-governance token. Otherwise leave it as `null`. ⚠

### Example.

Cake token paired with BUSD. Fill in **"intermediate\_token"** parameter in config.json file with BUSD address.

Pcv2 **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             |

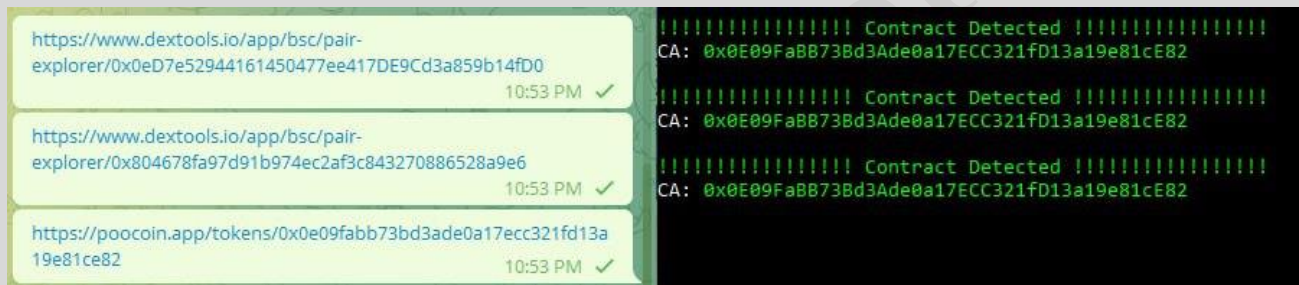
### Telegram Mode

⚠ Refer to **Config.json Setup** to obtain **"apiID"** and **"apiHash"** ⚠

⚠ Require telegram login information for first time use ⚠

⚠ Telegram scrapper is slower compare to other telegram bot due to more complex algorithm.

Increase gas to compensate for this ⚠



- ✓ Scan Telegram Public/Private Groups and Channels
- ✓ Convert letter number to decimal number such as *oNe* > 1, *Two* > 2, *ThReE* > 3
- ✓ Convert number emoji to number
- ✓ txt file > download > read
- ✓ Support basic arithmetic calculation
- ✓ Support blurred text in telegram chat
- ✓ Convert pair address to token address
- ✓ Remove garbage content and attempt to reconstruct to valid address
- ✓ Ignore greeting/bot messages.

### Chat Link Mode [1]

```
!!!!!! Please Select Telegram Chat Mode [Link/ID] !!!!!!!
Telegram Chat Link [1]:
Telegram Chat ID [2]:
1
Please Enter Telegram Chat Link [https://t.me/Example_Chat]:
https://t.me/PotatoSniperBot
!!!!!! Running !!!!!!!
```

This mode is only used in public group/channels where the link is provided. Simply copy and paste link into bot terminal as shown in diagram above. Right click on the terminal to paste and press ENTER.



## Tradable Mode

This mode will buy target token as soon as target contract is tradable and sellable and buy/sell tax is less than or equal to *“target buytax”* and *“target selltax”* parameter in config.json file.

⚠ INCREASE GAS TO BUY FASTER ⚠

This mode is only used in private group/channels where chat link is not provided. Simply copy and paste desired Chat ID into bot terminal as shown in diagram above. To copy, simply highlight the text and right click, right click again to paste into terminal and press ENTER.

## Set Fee (Mempool / Telegram / Tradable)

Enter buy amount and gas price, expected format **buy\_amount/gas\_price** or enter **0** to use default setting in config.json file.

### Example.

Buy 0.01 BNB with gas price of 5 gwei.

```
Enter Fee [Buy/Gas] or [0=Default]:  
0.01/5
```

## Limit Trade Mode

Uses "**contract\_owner\_addr**" for buy and sell

△ Maximize terminal, otherwise result will be spammed across the entire screen rather than printing on one line consistently △

```
!!!!!!!!!!!!!! Please Select Mode of Operation !!!!!!!!!!!!!!!  
Mempool Snipe [1]:  
Telegram Snipe [2]:  
Tradable Snipe [3]:  
Limit-Trade [4]:  
4  
  
!!!!!!!!!!!!!! Please Enter Contract Address !!!!!!!!!!!!!!!  
0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82  
  
!!!!!!!!!!!!!! My Balance !!!!!!!!!!!!!!!  
BNB  Cake:0  
  
!!!!!!!!!!!!!! Current LP Status !!!!!!!!!!!!!!!  
Price:9.112914212609011 Cake:15411902.959695855 WBNB:346017.69639485114 LP-Worth[BUSD]:140447349.52476326  
  
Warning: Liquidity Pool Too Small  
Please Enter Number of Price Target To Buy [$6.5/$6/$5.5 = 3] or [0]  
0  
Please Enter Number of Price Target To Sell [$6.5/$6/$5.5 = 3] or [0]  
0  
  
Current Configuration  
Block Chain: BSC  
Mode: Limit-Trade  
Router: 0x10ED43C718714eb63d5aA57B78B54704E256024E  
Target CA: 0x0E09Fa8B73Bd3Ade0a17ECC321fD13a19e81cE82  
Buy Price [BUSD]:  
Buy Amount [BNB]:  
Sell Price [BUSD]:  
Sell Amount [%]:  
StopLoss [BUSD]: null  
Gas: 0  
Gas Limit: 3000000  
  
Enter [Y] to continue / Enter [Space] to reset:  
y  
  
!!!!!!!!!!!!!! Running !!!!!!!!!!!!!!!  
  
Press [CTRL+C]: Terminate Script Any Time  
  
BNB  Cake:0.000 Price[BUSD]:9.11277224 Buy_Target:null Buy_Amount:null Sell_Target:null Sell_Amount:null Stop_Loss:null
```

Limit trade mode is used to buy and sell a specified token at specified single/multiple price targets and amounts. Checks for "**min\_bUSD\_worth\_of\_lp**" condition as shown in the diagram above.  
Supports governance/stable pair.

## Set Buy Target

```
Please Enter Number of Price Target To Buy [$6.5/$6/$5.5 = 3] or [0]
2
Please Enter Price-Target and Buy-Amount [Price in BUSD/ Buy Amount in BNB]
7.5/1.25
Please Enter Price-Target and Buy-Amount [Price in BUSD/ Buy Amount in BNB]
6/2.5
Please Enter Number of Price Target To Sell [$6.5/$6/$5.5 = 3] or [0]
0

Current Configuration
Block Chain: BSC
Mode: Limit-Trade
Router: 0x10ED43C718714eb63d5aA57B78B54704E256024E
Target CA: 0x0E09FaBB73Bd3Ade0a17ECC321fD13a19e81cE82
Buy Price [BUSD]: 7.5,6
Buy Amount [BNB]: 1.25,2.5
Sell Price [BUSD]:
Sell Amount [%]:
StopLoss [BUSD]: null
Gas: 0
Gas Limit: 3000000

Enter [Y] to continue / Enter [Space] to reset:
```

### Example.

Cake price = \$9, I want to buy when price is at \$7.5 and \$6 with amount of 1.25 BNB and 2.5 BNB.  
Verify configuration before proceeding.

### Set Sell Target

```
Please Enter Number of Price Target To Buy [$6.5/$6/$5.5 = 3] or [0]
0
Please Enter Number of Price Target To Sell [$6.5/$6/$5.5 = 3] or [0]
2
Please Enter Price-Target and Sell-Amount [Price in BUSD/ Sell Amount in %]
12/50
Please Enter Price-Target and Sell-Amount [Price in BUSD/ Sell Amount in %]
14/100
Please Enter Stop-Loss Price Target to Sell All Balance[Price in BUSD]
8

Current Configuration
Block Chain: BSC
Mode: Limit-Trade
Router: 0x10ED43C718714eb63d5aA57B78B54704E256024E
Target CA: 0x0E09FaBB73Bd3Ade0a17ECC321fD13a19e81cE82
Buy Price [BUSD]:
Buy Amount [BNB]:
Sell Price [BUSD]: 12,14
Sell Amount [%]: 50,100
StopLoss [BUSD]: 8
Gas: 0
Gas Limit: 3000000

Enter [Y] to continue / Enter [Space] to reset:
```

### Example.

Cake price = \$9, I want to sell when price is at \$12 and \$14 with amount of 50% and 100% of my total balance. If price of cake drops and goes below \$8 (*my stoploss price target*) then sell all balance. Verify configuration before proceeding.

## Config.json Setup

### api ID & Hash

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>

### url

Endpoint/Node url, rate limited Endpoint/Node not supported.

### block\_chain, router, factory, governance\_token\_addr

Refer to Address.txt, copy and paste from there

Example.

```
"block_chain": "BSC",
"router": "0x10ED43C718714eb63d5aA57B78B54704E256024E",
"factory": "0xcA143Ce32Fe78f1f7019d7d551a6402fC5350c73",
"governance_token_addr": "0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c",
"intermediate_token": null,
```

### Intermediate\_token

⚠ Fill in **"intermediate\_token"** parameter in config.json file, if target token is pairing with non-governance token. Otherwise leave it as **null**. ⚠

Example.

Cake token paired with BUSD. Fill in **"intermediate\_token"** parameter in config.json file with BUSD address.

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             |

### target\_owner

**target\_owner**

Leave It as is, bot will automatically fetch this. If bot failed to fetch owner address, then you need to manually enter it here. Only applies to **Mempool** Mode.

### my\_contract, hp\_contract

⚠ Ensure **Buy Contract** has enough to cover  $(buy\_amount * buy\_count + test\_buy\_amount)$  ⚠

**my\_contract** = Buy Contract

**hp\_contract** = Honeypot Contract

Obtain from ContractDeployer, Honeypot contract can be shared, if you can find someone who had already deployed the Honeypot Contract, you can use their contract instead of deploying your own.

### contract\_owner\_addr, contract\_owner\_key

**Warning:** ⚠ Only use test wallet and never use main wallet ⚠

⚠ Ensure **Contract Owner Wallet** has enough to cover gas fee ⚠

**contract\_owner\_addr** enter desire wallet address to deploy contracts.

**contract\_owner\_key** 64-character key

### sell\_account\_addr, sell\_account\_key

**Warning:** ⚠ Only use test wallet and never use main wallet ⚠

⚠ Ensure **Sell Wallet** has enough to cover gas fee ⚠

**sell\_account\_addr** enter desire wallet address for selling.

**sell\_account\_key** 64-character key

### auto\_sell config

**auto\_sell** Only accept *true* or *false*.

**auto\_sell\_interval** Sell interval at every x pure profit. Pure Profit = Current Balance - Investment Amount.

**auto\_sell\_amount** Sell x % at every sell interval.

**auto\_sell\_all\_target** Sell all balance at x pure profit.

### Telegram\_Honeypot, Telegram\_Download\_File

**Telegram\_Honeypot** Perform Honeypot check in *Telegram* mode. Only accept *true* or *false*.

**Telegram\_Download\_File** Enable download txt file in *Telegram* mode. Only accept *true* or *false*.

#### target\_buytax, target\_selltax

|                       |   |
|-----------------------|---|
| <b>target_buytax</b>  | Only buy if buy tax is $\leq x$ , only applies for <i>Telegram/Tradable</i> mode  |
| <b>target_selltax</b> | Only buy if sell Tax is $\leq x$ , only applies for <i>Telegram/Tradable</i> mode |

#### buy\_amount, buy\_count, test\_buy\_amount, buy\_delay\_timer, gas, gaslimit, sell\_slippage

|                        |  |
|------------------------|--|
| <b>buy_amount</b>      | Amount To Buy, Actual Buy                                  |
| <b>buy_count</b>       | How many times to buy within one transaction               |
| <b>test_buy_amount</b> | Amount To Buy, Sample Buy                                  |
| <b>buy_delay_timer</b> | Buy delay in seconds. Leave it as <i>null</i> if not used. |
| <b>gas</b>             | Gas price in gwei  |
| <b>gaslimit</b>        | Maximum allowed gas limit per transaction                  |
| <b>sell_slippage</b>   | Maximum acceptable difference $>$ expected vs actual.      |

#### limit\_trade

|                             |   |
|-----------------------------|---|
| <b>min_busd_worth_of_lp</b> | Minimum liquidity pool size in busd.                                    |
| <b>buy_slippage</b>         | Maximum acceptable difference $>$ expected vs actual.                   |
| <b>max_fail_attempts</b>    | Maximum fail transaction in a row before stopping all buy/sell.         |
| <b>approve_token</b>        | Approve token before trading? Only accept <i>true</i> or <i>false</i> . |