

Solana Memecoin Trading Bot

A customizable trading bot for Solana memecoins with social media scanning, token analysis, and automated trading.

Key Features

- Twitter scanning for memecoin discovery
- Token analysis with safety scoring
- Configurable trading parameters
- Dashboard for monitoring performance
- Simulation mode for risk-free testing

Quick Start

1. Set Up Environment

```
bash

# Create and activate a virtual environment
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install tweepy aiohttp solders solana-py python-dotenv pandas vaderSentiment streamlit plot
```

2. Configure API Keys

Create a `.env` file in the project directory:

```
# API Keys
TWITTER_BEARER_TOKEN=your_twitter_bearer_token
SOLANA_RPC_ENDPOINT=https://mainnet.helius-rpc.com/?api-key=your_api_key
WALLET_PRIVATE_KEY=your_wallet_private_key
DEXSCREENER_BASE_URL=https://api.dexscreener.com
JUPITER_QUOTE_API=https://quote-api.jup.ag/v6/quote
JUPITER_SWAP_API=https://quote-api.jup.ag/v6/swap
```

3. Run the Database Fix Script (First Time Only)

```
bash
```

```
python fix_database.py
```

4. Run the Bot

```
bash
```

```
python main.py
```

5. Run the Dashboard (Optional)

```
bash
```

```
streamlit run dashboard.py
```

Adjusting Trading Parameters

Option 1: Using the Dashboard

The dashboard provides an intuitive interface to adjust all trading parameters in real-time:

1. Launch the dashboard: `streamlit run dashboard.py`
2. Navigate to the Bot Controls section in the sidebar
3. Adjust parameters as needed
4. Click "Update Trading Parameters" to save changes

Option 2: Editing bot_control.json

You can manually edit `data/bot_control.json` to change trading parameters:

json

```
{
  "running": true,
  "simulation_mode": true,
  "max_investment_per_token": 1.0,
  "take_profit_target": 15.0,
  "stop_loss_percentage": 0.25,
  "slippage_tolerance": 0.30,
  "MIN_SAFETY_SCORE": 0.0,
  "MIN_VOLUME": 0.0,
  "MIN_LIQUIDITY": 0.0,
  "MIN_MCAP": 0.0,
  "MIN_HOLDERS": 0,
  "MIN_PRICE_CHANGE_1H": 0.0,
  "MIN_PRICE_CHANGE_6H": 0.0,
  "MIN_PRICE_CHANGE_24H": 0.0
}
```

Key Parameters Explained

Core Trading Parameters

- **Simulation Mode:** When enabled, trades are simulated without real execution
- **Max Investment per Token:** Maximum SOL to invest in a single token (default: 1.0 SOL)
- **Take Profit Target:** Multiplier for take profit (default: 15.0 = 1500% profit)
- **Stop Loss Percentage:** Percentage for stop loss trigger (default: 0.25 = 25%)
- **Slippage Tolerance:** Maximum allowed slippage percentage (default: 0.30 = 30%)

Token Screening Parameters

- **MIN_SAFETY_SCORE:** Minimum safety score (0-100) for token to be tradable
- **MIN_VOLUME:** Minimum 24-hour trading volume in USD
- **MIN_LIQUIDITY:** Minimum liquidity in USD
- **MIN_MCAP:** Minimum market capitalization in USD
- **MIN_HOLDERS:** Minimum number of token holders

Price Change Parameters

- **MIN_PRICE_CHANGE_1H:** Minimum price change over 1 hour
- **MIN_PRICE_CHANGE_6H:** Minimum price change over 6 hours

- **MIN_PRICE_CHANGE_24H:** Minimum price change over 24 hours

Starting with Very Lenient Parameters

For testing, we've set all minimum criteria to 0 to catch all possible tokens. As you refine your strategy, you can gradually increase these values to be more selective.

Troubleshooting

Common Issues

No Tokens Meeting Criteria

If you don't see any tokens qualifying for trading, lower the screening parameters:

- Set MIN_SAFETY_SCORE to 0
- Set MIN_VOLUME, MIN_LIQUIDITY, MIN_MCAP to 0
- Set MIN_HOLDERS to 0
- Set all MIN_PRICE_CHANGE values to 0

Database Errors

If you encounter database issues:

```
bash
```

```
python fix_database.py --reset
```

API Rate Limiting

The bot includes automatic rate limiting mechanisms, but if you encounter API rate limit errors, increase the scan interval in config.py.

Important Notes

- Always start in simulation mode
- Monitor the logs for any errors
- Use the dashboard to visualize performance
- The bot is set to a 15x profit target by default

Happy trading! 🚀