🔧 Key Components Explained

1. Dependencies (need to load.txt)

Python libraries required to run the scripts:

- requests: For API calls
- web3, tronpy: For blockchain interaction
- json, time, sys: Utility modules

Install with:

```
pip install requests web3 tronpy
```

---

2. Multi-Chain Tracker (TRON_ERC_CCHAIN_BEP.py)

Monitors:

- Ethereum (ERC20)
- Binance Smart Chain (BEP20)
- Avalanche (C-Chain)
- TRON (TRC20)

🔍 Key Features:

- Uses RPC nodes/APIs (Etherscan, TronGrid)
- Scans predefined tokens (USDT, ETH, BNB, etc.)
- Aggregates balances across addresses

📄 Sample Output:

```
🔷 Scanning address: 0xD3C3...CcB
 🌋 Avalanche: AVAX: 1.205
 🔷 Ethereum: ETH: 0.54, USDT: 1200.0
 🍌 BSC: CAKE: 85.2
```

---

3. Solana Tracker (SOL.py)

- Retrieves SOL balances via Solana's JSON-RPC API
- Converts lamports to SOL (1 SOL = 1 billion lamports)

---

4. UTXO Coin Tracker (BTC_LTC_DOGE.py)

- Uses BlockCypher API to check balances for:
  - Bitcoin (BTC)
  - Litecoin (LTC)
  - Dogecoin (DOGE)
- Manages rate limits and retries

---

5. Algorand Tracker (ALGO.py)

- Retrieves ALGO balances via AlgoNode API
- Converts microAlgos to ALGO (1 ALGO = 1 million microAlgos)

---

6. Address / Token Reference (adres to py.xlsx)

- Sheet 1: Formats wallet addresses into Python lists
- Sheet 2: Master token list with decimals for:
  - Ethereum (ERC20)
  - Binance Smart Chain (BEP20)
  - Avalanche (C-Chain)
  - TRON
  - Example entry: ("USDT", "0xdac17...", 6)

🔁 How It Works Together

Input: Wallet addresses defined in each script (e.g., ADDRESSES inside TRON_ERC_CCHAIN_BEP.py)
Execution: Scripts query blockchain nodes or APIs
Output:

- Individual token balances
- Chain-specific totals
- Aggregated balances across all addresses

📊 Sample Output:

```
🌋 AVAX Totals: USDC: 4500.0
🔷 ETH Totals: ETH: 2.1
🍌 BSC Totals: CAKE: 210.5
```

💼 Use Cases

- **Portfolio Monitoring:** Unified balance overview across chains
- **Tax Reporting:** Exportable snapshots for documentation
- **Wallet Auditing:** Validate expected balances
- **Fund Tracking:** Monitor treasury or DAO activity

🚀 How to Run

1. Install dependencies listed in need to load.txt
2. Populate wallet addresses in each script
3. Run scripts individually:

```
python TRON_ERC_CCHAIN_BEP.py
python SOL.py
python BTC_LTC_DOGE.py
python ALGO.py
```

🔑 Note: Replace any placeholder API keys (e.g., ETHERSCAN_API_KEY) with your own for reliability

İbrahim Tolga TOPSAKAL