Options Chain Data Processing

Overview:

This project retrieves and processes options chain data for financial instruments like **NIFTY** and **BANKNIFTY**. The goal is to extract the highest bid for put options (PE) and the highest ask for call options (CE), and calculate margin requirements and premiums earned for each option contract.

Project Structure:

- **get_option_chain_data_upstox**: Fetches option chain data from the Upstox API or mock data file.
- **calculate_margin_and_premium_upstox**: Calculates the margin requirements and premiums based on retrieved data.
- **Mock and Synthetic Data**: Included to simulate real-world scenarios if API access is unavailable.

Approach to API Unavailability:

Due to potential access restrictions with the Upstox API, I utilized mock data and synthetic data generation to simulate realistic options chain data:

1. **Mock Data**: A JSON file (option_chain_data.json) was created to mirror the structure of a typical API response.
2. **Synthetic Data Generation**: A function was added to generate random but realistic options chain entries, providing flexibility in testing and validation.

Usage:

1. **API Fetching**:

python

Copy code

df = get_option_chain_data_upstox(api_key, 'NSE_INDEX|Nifty 50', '2024-03-28', 'PE')

2. **Mock Data Loading**:

python

Copy code

df = load_mock_data('option_chain_data.json', 'PE')

3. **Synthetic Data Generation**:

python

Copy code

synthetic_df = generate_synthetic_data(10, 'NIFTY', 'CE')

Requirements:

- Python 3.8+
- Packages: requests, pandas, numpy (for synthetic data generation)

Future Integration:

The code structure allows for easy switching between mock data and live API data, making it adaptable for real-world use once API access becomes available.