

## Battery Trading Bot (Romania PZU and Balancing)

### Overview

- Automated trading framework for a 55 MWh battery targeting OPCOM PZU (Day-Ahead) and Balancing markets.
- Includes dry-run and backtest modes; live mode requires implementing exchange adapters and market access.
- Focuses on modularity: market clients, strategy, risk, execution, and scheduling.

### Legal and compliance

- Operating on PZU/Balancing requires valid market access, registrations, and compliance with Romanian/EU regulations (e.g., OPCOM/Transelectrica rules, REMIT/ACER).
- This software is provided as a template. Validate all integrations, limits, and procedures before live trading.

### Quick start

1. Prerequisites: Python 3.11+, macOS/Linux.
2. Create env and install deps:  

```
python -m venv .venv && source .venv/bin/activate  
pip install -r requirements.txt
```
3. Configure environment:  

```
cp .env.example .env
```

  
Edit config.yaml (battery, risk, data files, execution mode) and .env (credentials/endpoints).
4. Run (dry-run by default):  

```
python -m src.main --mode dry-run --date YYYY-MM-DD
```

  
Backtest (no API calls):  

```
python -m src.main --mode backtest --date YYYY-MM-DD
```

  
Live (placeholders; implement clients first):  

```
python -m src.main --mode live --date YYYY-MM-DD
```

### Project structure

- src/market: Market client base and OPCOM PZU/Balancing stubs.
- src/strategy: Battery strategy to create orders from price forecasts.
- src/risk: SOC and risk controls (price/volume caps).
- src/execution: Order routing to clients.
- src/data: Price forecast loaders (CSV/API hooks).
- src/scheduling: Daily/interval scheduling hooks.
- src/main.py: Entrypoint and CLI.

### Notes

- Price data access and order routing require official APIs or integration with your provider/BRP. Replace stubs with real endpoints.
- Timezone defaults to Europe/Bucharest.

- Use dry-run/backtest to validate logic and limits before live.

#### Automated downloads (optional)

- Requires Playwright browsers: `pip install -r requirements.txt && python -m playwright install chromium`
- Run auto-download + aggregation:  
`RUN_PLAYWRIGHT=1 YEARS=1 ./download_and_aggregate.sh 4.97`
- Outputs daily CSVs under `downloads/*/auto` and aggregated CSVs under `data/`
- If site structure changes, adjust selectors in `src/tools/download_*.py`