EMA “Captain” Crypto Trading Bot (with RSI/MACD “Advisors”)

This bot watches multiple crypto markets on Coinbase in real time. It makes simple, explainable trades:

- **EMA crossover** is the “captain”: when a short-term moving average crosses a long-term one, that’s a buy/sell signal.
- **RSI & MACD** are “advisors”: they can veto obviously bad buys/sells.
- Orders are **post-only limit orders** by default (to earn maker rates), with a **daily spend cap** and a **cool-down** between trades per coin. It keeps a lightweight portfolio and P&L.

Quick start (plain English)

1. **Add your API keys** in an env file (default name APIkeys.env) or environment variables:
    - COINBASE_API_KEY, COINBASE_API_SECRET, and optional COINBASE_PORTFOLIO_ID.
2. **Run the bot**:
3. python main.py

It sets up logging, loads keys, starts the bot, and idles until you stop it. Ctrl-C triggers a clean shutdown.

What each file does (in friendly terms)

- **main.py** – The entry point. Loads your keys, wires up Ctrl-C handling, starts the bot, and keeps the event loop alive until you stop it.
- **bot/config.py** – All the knobs. Pick which coins to trade, EMA lengths, spend limits, cool-down, advisor (RSI/MACD) settings, and maker offsets. Includes per-coin overrides and an EMA “dead-band” to avoid flapping.
- **bot/tradebot.py** – The brain. Connects to Coinbase (REST + WebSocket), subscribes to tickers for all chosen coins, updates indicators per tick, decides when to trade, enforces daily cap/cool-down, places orders (maker by default), and updates portfolio/P&L from fills. Also logs a session P&L footer on exit.
- **bot/indicators.py** – Lightweight, streaming indicators:

- **EMA** (short & long) for the captain signal,
- **RSI(14)** by default,
- **MACD(12,26,9)** with signal line & histogram.

- **bot/strategy.py** – Advisor rules. RSI must be within a safe range; MACD must not contradict the side (default: buy only if MACD≥0, sell only if MACD≤0). It can normalize MACD by price (basis points).
- **bot/orders.py** – Builds post-only limit orders near the current bid/ask with an offset, and rounds price/size to the exchange increments. Returns the price & size to place.
- **bot/persistence.py** – The little database:
  - **SpendTracker** (daily spend cap),
  - **LastTradeTracker** (per-coin cool-down),
  - **PortfolioStore** (positions, cost basis, realized P&L),
  - **ProcessedFills** (avoid double-counting),
  - Trade log writer (appends human-readable lines).
- **bot/constants.py** – Points the bot to a .state/ folder and names the JSON/text files it uses.
- **bot/logging_setup.py** – Optional rotating file logs; sets formats and quiets noisy libraries.
- **bot/utils.py** – Thin re-export layer so tradebot.py can import the persistence & constants helpers from one place.

How the bot decides to trade (simple view)

1. **Real-time prices** stream in for every chosen product.
2. For each tick, it updates **EMA/RSI/MACD** for that product. It waits for a minimum warm-up of ticks first.
3. When **short EMA crosses long EMA** (with a small “dead-band” buffer), that’s a new buy/sell event.
4. **Advisors can veto** if RSI is out of range or MACD disagrees with the side.

5. It **obeys cool-down** (per coin) and **daily spend cap**; if you hit the cap, it logs P&L and **stops the session**.
6. **Orders**: by default it places **post-only limit** buys/sells using your maker offset (bps) relative to bid/ask; market orders are optional.

Key settings you'll actually tweak

All of these live in **bot/config.py** unless noted.

What to trade & how much

- product_ids: the coins you want. More coins = more potential trades overall.
- usd_per_order: size of each order in USD.
- max_usd_per_day: daily spend cap; hitting it will stop the bot for the day.
- dry_run: simulate without sending live orders.

Signal sensitivity (frequency dial)

- short_ema, long_ema: shorter numbers react faster (more trades); longer numbers react slower (fewer). Per-coin overrides available in ema_params_per_product.
- ema_deadband_bps: extra buffer around crossovers to avoid choppy flip-flops. Bigger dead-band = fewer signals.
- cooldown_sec: minimum seconds between trades **per coin**. Higher = fewer trades.
- min_ticks (and per-coin min_ticks in overrides): warm-up period (doesn't change steady-state frequency, just how soon it can start).

Advisor (RSI/MACD) guardrails

- Toggle with enable_advisors / use_advisors. Turning them off increases trades.
- **RSI band**: rsi_buy_floor (default 50) and rsi_sell_ceiling (default 70). Buy/Sell only allowed when RSI is within this band; narrowing the band reduces trades, widening increases.
- **MACD sign-check**: by default, buys need MACD ≥ 0 and sells need ≤ 0. (Advanced: you can loosen/tighten thresholds in code: strategy.AdvisorSettings.macd_buy_min / macd_sell_max.)

Order style (fill behavior)

- prefer_maker: True = post-only limit orders.
- maker_offset_bps (and per-coin map): bigger offset = less aggressive pricing, **fewer fills**; smaller offset = more fills.

Sell guard (risk)

- max_loss_bps: don't sell below cost basis beyond this tolerance (bps). Lower value = fewer loss-taking sells; higher value = more sells allowed when under water.

Tuning cheatsheet

If you're seeing *too many* trades per session:

- **Raise cooldown_sec** (e.g., 720 → 1200).
- **Increase ema_deadband_bps** (e.g., 8 → 12–20 bps).
- **Lengthen EMAs** (e.g., short_ema +10–30%, long_ema +10–30%). Use per-coin overrides if only some assets are noisy.
- **Narrow the RSI window** (raise rsi_buy_floor, lower rsi_sell_ceiling, e.g., 55–65).
- **Increase maker_offset_bps** so limits are less likely to fill in chop.
- **Lower max_usd_per_day** to force the bot to stop sooner once busy.

If you're getting *hardly any* trades:

- **Lower cooldown_sec** (e.g., 720 → 300).
- **Shrink ema_deadband_bps** (e.g., 8 → 4–0 bps).
- **Shorten EMAs** (e.g., short_ema −10–30%, long_ema −10–30%).
- **Widen the RSI window** (lower rsi_buy_floor, raise rsi_sell_ceiling, e.g., 45–75), or **disable advisors** temporarily.
- **Decrease maker_offset_bps** for more fills, or set prefer_maker=False (market orders).
- **Raise max_usd_per_day** if you're hitting the cap early.

Advanced MACD loosen/tighten: In code, the default requires **MACD ≥ 0 for buys** and **≤ 0 for sells**. To make it less strict (more trades), allow small negatives/positives by changing macd_buy_min and macd_sell_max when AdvisorSettings is created in tradebot.py. (Example: set them to −5/+5 bps. macd_buy_min=-5, macd_sell_max=+5)

Where things get saved

- **State folder**: .state/ (configurable via BOT_STATE_DIR).
- **Files**:
  daily_spend.json, last_trades.json, trade_log.txt, portfolio.json, processed_fills.json.
- **Trade log lines** look like: timestamp, side, product, USD, price, qty (and "DRY RUN" if simulated).
- **Session P&L footer** is appended when the session ends.

Order placement (maker vs market)

- **Maker (default):** prices are placed slightly *away* from current bid/ask using your offset (in basis points). This can reduce fees but may not fill in fast markets.
- **Market:** switch by setting prefer_maker=False in config.

Notes & defaults (handy)

- **RSI default period**: 14. **MACD**: 12/26 with 9-period signal.
- **Dead-band meaning**: a small % buffer (in **bps**, where 100 bps = 1%) around the crossover to cut down on noise.
- **Sell guard**: won't sell at a loss beyond max_loss_bps below cost basis unless you relax that number.