# Binance Trading Bot — MVP (3-Hour Build)

Status: Paper-trading by default (safe). Flip `LIVE_TRADING=true` *after* testing and adding real API keys (Binance or Binance.US).
Stack: FastAPI, Python, SQLite (via SQLAlchemy), simple EMA crossover strategy, Docker or local venv.

## 1) Quickstart — Local (with helper script)

```
chmod +x run_bot.sh
./run_bot.sh
```

This script will:

1. Create `.env` from `.env.example` if missing.
2. Automatically change the API base URL to Binance.US (`https://api.binance.us`).
3. Activate your `.venv`.
4. Start the bot at `http://localhost:8010/docs`.

## 2) Quickstart — Docker

```
# from inside the unzipped project folder
cp .env.example .env
# Update BINANCE_BASE_URL to https://api.binance.us if in US
docker compose up --build
```

Open API docs: http://localhost:8000/docs

## 3) Quickstart — Manual Local venv

```
python3 -m venv .venv
source .venv/bin/activate   # Windows: .venv\Scripts\activate
pip install --upgrade pip
pip install -r requirements.txt

cp .env.example .env
# IMPORTANT: Edit .env to set:
# BINANCE_BASE_URL=https://api.binance.us
uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
```

## 4) What it does

- Pulls recent candles from Binance public API (no key needed).
- Computes EMA(9) vs EMA(21) crossover.
- Paper broker simulates market orders with configurable fee & slippage.
- Stores trades and positions in `trades.db` (SQLite).
- FastAPI endpoints to run strategy on demand, inspect trades, and reset state.

Live trading is scaffolded. Turn on with `LIVE_TRADING=true` and set keys. Use at your own risk.

## 5) Environment (.env)

```
# Core
SYMBOL=BTCUSDT
INTERVAL=1m
LOOKBACK=300

# Paper broker
PAPER_STARTING_USDT=10000
PAPER_FEE_RATE=0.001        # 0.1%
PAPER_SLIPPAGE_BPS=5        # 5 bps = 0.05%

# Strategy
FAST_EMA=9
SLOW_EMA=21
POSITION_SIZE_USDT=250      # per BUY in paper mode

# Live trading (off by default)
LIVE_TRADING=false
BINANCE_API_KEY=
BINANCE_API_SECRET=
BINANCE_BASE_URL=https://api.binance.com      # For Binance.US, set: https://api.binance.us
```

## 6) API Endpoints (http://localhost:8000/docs)

- `GET /health` — sanity check
- `POST /strategy/run` — fetch candles, compute signal, place paper/live order if needed
- `GET /trades` — list saved trades
- `POST /paper/reset` — reset the paper account and wipe trades/positions

## 7) Notes & Next Steps

- Add background scheduler to run every N minutes.
- Add risk controls (max daily loss, cooldowns, position limits).
- Add more strategies (RSI, MACD, BBands) and portfolio router.
- Wire **live** order execution via REST (currently scaffolded & safe by default).
- Integrate Discord/Slack alerts for fills and PnL.