FXOpen AI Trading Bot - main.py

```python
from failsafe import *  # this sets the global exception hook automatically

import os
import time
import hmac
import json
import hashlib
import logging
import asyncio
from datetime import datetime
from dotenv import load_dotenv
from telegram import Bot

# Your bot modules (adjust imports if you changed file names)
from ai_analyzer import AIAnalyzer
from earnings_tracker import EarningsTracker
from screenshot import Screenshot
from trader import Trader  # This Trader should be rewritten for FXOpen already
from telegram_bot import TelegramBot

# Logging setup
logging.basicConfig(level=logging.INFO, format="%(asctime)s - %(message)s")

# Load env variables
load_dotenv()
API_KEY = os.getenv("OPENAI_API_KEY")
TG_TOKEN = os.getenv("TELEGRAM_BOT_TOKEN")
TG_CHAT_ID = os.getenv("TELEGRAM_CHAT_ID")
FXOPEN_LOGIN = os.getenv("FXOPEN_LOGIN")
FXOPEN_KEY = os.getenv("FXOPEN_KEY")
FXOPEN_SECRET = os.getenv("FXOPEN_SECRET")

BASE_URL = "https://restapi.fxopen.com:8443"

# Telegram Bot instance for direct sends outside TelegramBot class
bot = Bot(token=TG_TOKEN)



def hmac_auth(endpoint, payload=None):
    nonce = str(int(time.time() * 1000))
    body = json.dumps(payload) if payload else ""
    message = nonce + endpoint + body
    signature = hmac.new(
        FXOPEN_SECRET.encode("utf-8"), message.encode("utf-8"), hashlib.sha256
    ).hexdigest()
    headers = {
        "X-Auth-Apikey": FXOPEN_KEY,
        "X-Auth-Nonce": nonce,
        "X-Auth-Signature": signature,
        "Content-Type": "application/json",
    }
    return headers


def get_account_info():
    endpoint = f"/accounts/{FXOPEN_LOGIN}"
    url = BASE_URL + endpoint
    headers = hmac_auth(endpoint)
    r = requests.get(url, headers=headers)
    if r.status_code == 200:
        return r.json()
    else:
        logging.error(
            f"FXOpen account info fetch failed: {r.status_code} {r.text}")
        return {}


async def send_update():
    account = get_account_info()
    balance = account.get("Balance", "N/A")
    equity = account.get("Equity", "N/A")
    margin = account.get("UsedMargin", "N/A")

    msg = (
        f" FXOpen Bot Report:\n"
        f"Balance: ${balance}\n"
        f"Equity: ${equity}\n"
        f"Used Margin: ${margin}\n"
        f"Time: {datetime.utcnow().strftime('%H:%M:%S UTC')}"
    )
    try:
        await bot.send_message(chat_id=TG_CHAT_ID, text=msg)
        logging.info("Update sent.")
    except Exception as e:
        logging.error(f"Failed to send Telegram update: {e}")


async def periodic_update(interval_sec=60):
    while True:
        await send_update()
        await asyncio.sleep(interval_sec)


async def market_scanner(trader):
    while True:
        try:
            await trader.evaluate_and_execute()


async def main():
    load_dotenv()

    # Initialize modules
    ai = AIAnalyzer()
    earnings = EarningsTracker()
    ss = Screenshot()

    # Set up Telegram bot and trader
    bot = TelegramBot(None, earnings)
    trader = Trader(ai, earnings, ss, bot)
    bot.trader = trader

    # Start background scanner
    asyncio.create_task(market_scanner(trader))

    # Send first update
    await send_update()

    # Start bot polling
    await bot.start_polling()

if __name__ == "__main__":
    logging.info("FXOpen AI Bot running.")
```

```
asyncio.run(main())
```