# **NEWS SCOUT BOT - COMPLETE SETUP**

## **STEP 1: ADD TELEGRAM SECRETS TO GITHUB**

1. Go to:
2. <https://github.com/reindolfpratt/TRADING-MODEL>
3. Click Settings → Secrets and variables → Actions
4. Click "New repository secret"
5. First Secret:
   * Name: TELEGRAM\_BOT\_TOKEN
   * Secret: 8223831843:AAF5bTpLNfJwG5wJSXm71ep8eBIjWhmp4IQ
   * Click "Add secret"
6. Click "New repository secret" again
7. Second Secret:
   * Name: TELEGRAM\_CHAT\_ID
   * Secret: 7659687185
   * Click "Add secret"

✅ You should now see 4 secrets total:

* EMAIL\_USER
* EMAIL\_PASSWORD
* TELEGRAM\_BOT\_TOKEN
* TELEGRAM\_CHAT\_ID

## **STEP 2: CREATE THE BOT FILE**

1. Go back to repo main page
2. Click "Add file" → "Create new file"
3. Filename: news\_scout\_bot.py
4. Copy and paste this COMPLETE code:

import feedparser

import yfinance as yf

from datetime import datetime, timedelta

import time

import requests

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

import os

# ═══════════════════════════════════════════════════════════

# CONFIGURATION

# ═══════════════════════════════════════════════════════════

TELEGRAM\_BOT\_TOKEN = os.environ.get('TELEGRAM\_BOT\_TOKEN')

TELEGRAM\_CHAT\_ID = os.environ.get('TELEGRAM\_CHAT\_ID')

# Stocks to monitor (15 most volatile)

STOCKS\_TO\_MONITOR = [

"AAPL", "TSLA", "NVDA", "AMZN", "GOOGL",

"MSFT", "META", "NFLX", "AMD", "COIN",

"SPY", "QQQ", "DIS", "PLTR", "SOFI"

]

# Initialize sentiment analyzer

analyzer = SentimentIntensityAnalyzer()

# Track recently alerted stocks (prevent duplicates)

recently\_alerted = {}

COOLDOWN\_HOURS = 2 # Don't alert same stock for 2 hours

# ═══════════════════════════════════════════════════════════

# FUNCTIONS

# ═══════════════════════════════════════════════════════════

def get\_latest\_news\_rss(symbol):

"""Get latest news from Yahoo Finance RSS feed"""

try:

url = f"https://feeds.finance.yahoo.com/rss/2.0/headline?s={symbol}&region=US&lang=en-US"

feed = feedparser.parse(url)

articles = []

for entry in feed.entries[:3]:

articles.append({

'title': entry.title,

'link': entry.link

})

return articles

except Exception as e:

print(f"Error fetching RSS for {symbol}: {e}")

return []

def analyze\_sentiment\_vader(headlines):

"""Analyze sentiment using VADER AI"""

if not headlines:

return "NEUTRAL", 0, "No news available"

text = " ".join([h['title'] for h in headlines])

scores = analyzer.polarity\_scores(text)

compound = scores['compound']

if compound >= 0.5:

sentiment = "BULLISH"

impact = min(10, int((compound - 0.5) \* 20) + 6)

elif compound <= -0.5:

sentiment = "BEARISH"

impact = min(10, int((-compound - 0.5) \* 20) + 6)

elif compound >= 0.2:

sentiment = "BULLISH"

impact = 5

elif compound <= -0.2:

sentiment = "BEARISH"

impact = 5

else:

sentiment = "NEUTRAL"

impact = 3

reasoning = headlines[0]['title'][:100] if headlines else "No specific news"

return sentiment, impact, reasoning

def get\_current\_price(symbol):

"""Get current stock price"""

try:

ticker = yf.Ticker(symbol)

hist = ticker.history(period='1d', interval='5m')

if len(hist) < 1:

return None

current\_price = float(hist['Close'].iloc[-1])

return round(current\_price, 2)

except Exception as e:

print(f"Price error for {symbol}: {e}")

return None

def send\_telegram\_alert(symbol, action, price, impact, reasoning):

"""Send instant Telegram alert with trade details"""

if not TELEGRAM\_BOT\_TOKEN or not TELEGRAM\_CHAT\_ID:

print(f"⚠️ Telegram not configured - Would send: {symbol} {action}")

return

if "BUY LONG" in action:

target1 = round(price \* 1.03, 2)

target2 = round(price \* 1.05, 2)

stop = round(price \* 0.985, 2)

message = f"""🚨 \*TRADE ALERT\* 🚨

{action} \*{symbol}\*

💵 Entry Price: ${price}

🎯 \*PROFIT TARGETS:\*

• Take Profit 1: ${target1} (+3%)

• Take Profit 2: ${target2} (+5%)

🛑 \*STOP LOSS:\* ${stop} (-1.5%)

📊 Impact: {impact}/10

💡 \_{reasoning}\_

⏰ {datetime.now().strftime('%H:%M:%S')}"""

else:

target1 = round(price \* 0.97, 2)

target2 = round(price \* 0.95, 2)

stop = round(price \* 1.015, 2)

message = f"""🚨 \*TRADE ALERT\* 🚨

{action} \*{symbol}\*

💵 Entry Price: ${price}

🎯 \*PROFIT TARGETS:\*

• Cover at 1: ${target1} (-3%)

• Cover at 2: ${target2} (-5%)

🛑 \*STOP LOSS:\* ${stop} (+1.5%)

📊 Impact: {impact}/10

💡 \_{reasoning}\_

⏰ {datetime.now().strftime('%H:%M:%S')}"""

url = f"https://api.telegram.org/bot{TELEGRAM\_BOT\_TOKEN}/sendMessage"

payload = {

'chat\_id': TELEGRAM\_CHAT\_ID,

'text': message,

'parse\_mode': 'Markdown'

}

try:

response = requests.post(url, json=payload, timeout=10)

if response.status\_code == 200:

print(f"✅ Telegram alert sent for {symbol}")

else:

print(f"❌ Telegram failed: {response.text}")

except Exception as e:

print(f"❌ Telegram error: {e}")

def scan\_all\_stocks():

"""Main scanner - checks all stocks for news opportunities"""

print(f"\n🔍 Starting market scan at {datetime.now().strftime('%H:%M:%S')}")

print(f"📊 Monitoring {len(STOCKS\_TO\_MONITOR)} stocks for breaking news...\n")

opportunities\_found = 0

for symbol in STOCKS\_TO\_MONITOR:

print(f"Scanning {symbol}...", end=" ")

news = get\_latest\_news\_rss(symbol)

if not news:

print("No news")

continue

sentiment, impact, reasoning = analyze\_sentiment\_vader(news)

print(f"{sentiment} ({impact}/10)", end="")

# Only alert on impact 9+ (HUGE news only)

if impact >= 9:

# Check cooldown - don't spam same stock

now = datetime.now()

if symbol in recently\_alerted:

last\_alert\_time = recently\_alerted[symbol]

time\_since\_alert = (now - last\_alert\_time).total\_seconds() / 3600

if time\_since\_alert < COOLDOWN\_HOURS:

print(f" (Already alerted {time\_since\_alert:.1f}h ago - skipping)")

continue

price = get\_current\_price(symbol)

if price:

print(f" 🚨 HIGH IMPACT - Sending alert!")

if sentiment == "BULLISH":

action = "🟢 BUY LONG"

else:

action = "🔴 SHORT"

send\_telegram\_alert(symbol, action, price, impact, reasoning)

# Mark as alerted

recently\_alerted[symbol] = now

opportunities\_found += 1

time.sleep(2)

else:

print(" (Price unavailable)")

else:

print()

time.sleep(0.5)

print(f"\n✅ Scan complete! Found {opportunities\_found} high-impact opportunities.")

print(f"⏰ Next scan in 10 minutes\n")

# ═══════════════════════════════════════════════════════════

# RUN THE SCANNER

# ═══════════════════════════════════════════════════════════

if \_\_name\_\_ == "\_\_main\_\_":

scan\_all\_stocks()

1. Click "Commit new file"

## **STEP 3: CREATE THE WORKFLOW**

1. Click "Add file" → "Create new file"
2. Filename: .github/workflows/news-scout.yml
3. Copy and paste this:

text

name: News Scout Bot

on:

schedule:

- cron: '\*/10 13-20 \* \* 1-5' # Every 10 min, 1:30PM-8PM UTC, Mon-Fri

workflow\_dispatch: # Manual trigger button

jobs:

scout:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v3

- name: Set up Python

uses: actions/setup-python@v4

with:

python-version: '3.11'

- name: Install dependencies

run: |

pip install yfinance feedparser vaderSentiment requests

- name: Run news scout

env:

TELEGRAM\_BOT\_TOKEN: ${{ secrets.TELEGRAM\_BOT\_TOKEN }}

TELEGRAM\_CHAT\_ID: ${{ secrets.TELEGRAM\_CHAT\_ID }}

run: python news\_scout\_bot.py

1. Click "Commit new file"

## **STEP 4: ACTIVATE YOUR BOT**

## **Use Direct Link**

Open this link in your phone's browser:

<https://t.me/reindolf_trading_bot>

It will say "Open in Telegram" - Click it

Click "START" button

Send message: hello

This lets the bot know it can send you messages!

## **STEP 5: TEST IT NOW!**

1. Go to Actions tab in your repo
2. Click "News Scout Bot" (left sidebar)
3. Click "Run workflow" dropdown (right side)
4. Click green "Run workflow" button
5. Wait 1-2 minutes
6. Click on the workflow run to watch it
7. Click "scout" → "Run news scout" to see logs
8. Check your Telegram!

## **WHAT TO EXPECT:**

## **If high-impact news exists:**

✅ You'll get Telegram alert with trade details

## **If no high-impact news:**

✅ Workflow runs successfully  
✅ Logs say "Found 0 high-impact opportunities"  
✅ This is NORMAL - most scans find nothing (which is good!)

## **VERIFICATION:**

Your repo should now have:

* ✅ signal.py (daily email bot)
* ✅ news\_scout\_bot.py (news alert bot)
* ✅ .github/workflows/daily-signal.yml (daily email schedule)
* ✅ .github/workflows/news-scout.yml (news scout schedule)

Your secrets should have:

* ✅ EMAIL\_USER
* ✅ EMAIL\_PASSWORD
* ✅ TELEGRAM\_BOT\_TOKEN
* ✅ TELEGRAM\_CHAT\_ID

NOW RUN THE TEST! Go to Actions → News Scout Bot → Run workflow! 🚀