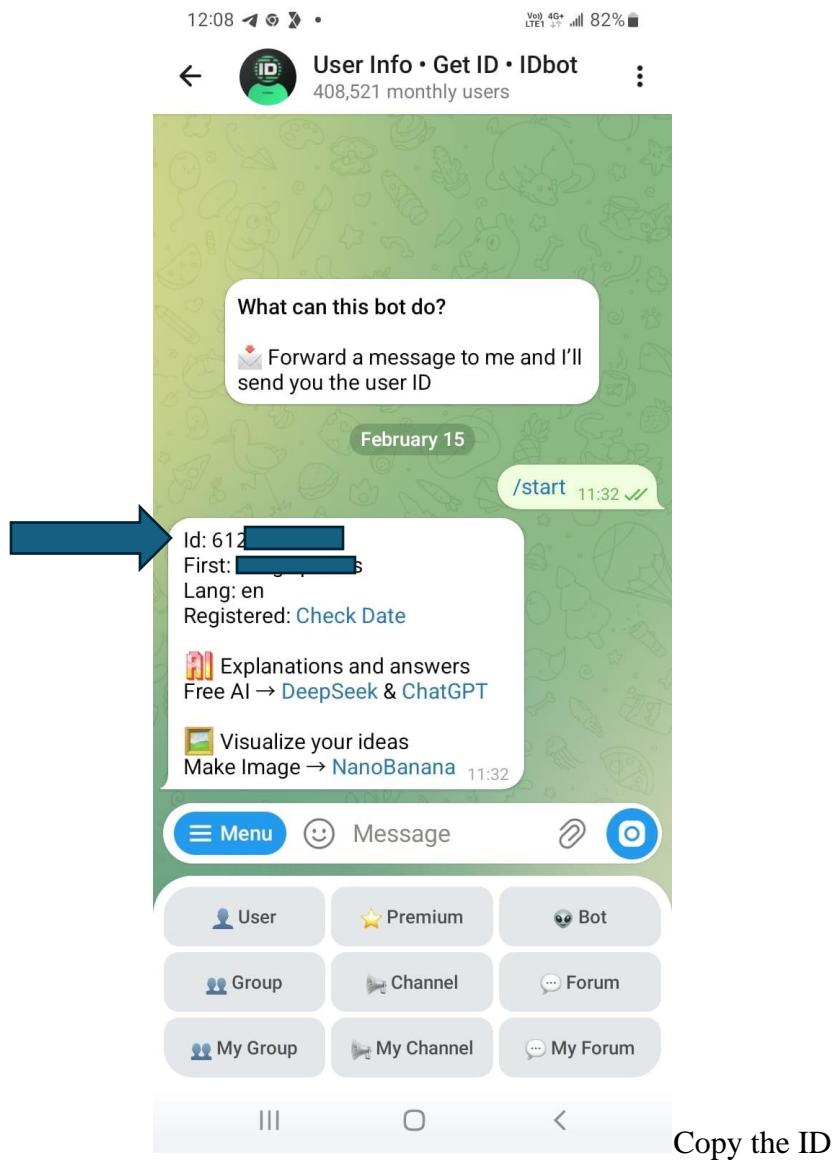


In telegram search box, type@botfather, then /newbot, follow the steps and obtain the token (see the arrows).



Copy the ID

https://github.com/econ105/AI/blob/main/Python/Algotrading/telegram/performance_chart.ipynb

Copy your own token ID to the Python code

[Open in Colab](#)

```
In [4]: !pip install python-telegram-bot --upgrade

Requirement already satisfied: python-telegram-bot in /usr/local/lib/python3.12/dist-packages (22.6)
Requirement already satisfied: httpx<0.29,>=0.27 in /usr/local/lib/python3.12/dist-packages (from python-telegram-bot) (0.28.1)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<0.29,>=0.27->python-telegram-bot) (4.12.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<0.29,>=0.27->python-telegram-bot) (2026.1.4)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<0.29,>=0.27->python-telegram-bot) (1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<0.29,>=0.27->python-telegram-bot) (3.11)
Requirement already satisfied: h11>0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<0.29,>=0.27->python-telegram-bot) (0.16.0)
Requirement already satisfied: typing_extensions>=4.5 in /usr/local/lib/python3.12/dist-packages (from anyio->httpx<0.29,>=0.27->python-telegram-bot) (4.15.0)

In [5]: import asyncio
        from telegram import Bot

        TELEGRAM_BOT_TOKEN = "REDACTED"
        TELEGRAM_CHAT_ID = "REDACTED" # your chat id

        async def main():
            bot = Bot(token=TELEGRAM_BOT_TOKEN)
            await bot.send_message(chat_id=TELEGRAM_CHAT_ID, text="hi this is testing")

        # Jupyter-safe way to run the coroutine
        loop = asyncio.get_event_loop()
        if loop.is_running():
            # For Jupyter/IPython: schedule and wait
            task = loop.create_task(main())
        else:
            loop.run_until_complete(main())
```

In [6]:

```
import asyncio
from datetime import datetime, timedelta

import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
from telegram import Bot

TELEGRAM_BOT_TOKEN = "1666447235:AHHGohNjqq-PbDlK1G-SJTigQbgU7D8w9jU"
TELEGRAM_CHAT_ID = "612618566" # use integer, your chat id

TICKERS = ["HL", "SLV"]
PERIOD_DAYS = 60 # ~ Last 2 months
CHART_FILENAME = "rel_perf_HL_SLV.png"

def download_data(tickers, period_days=60):
    end = datetime.today()
    start = end - timedelta(days=period_days)
    data = yf.download(
        tickers=tickers,
        start=start,
        end=end,
        interval="1d",
        auto_adjust=False,
        progress=False,
    )
    # Handle single vs multi-index columns
    if isinstance(data.columns, pd.MultiIndex):
        close = data["Close"].copy()
    else:
        close = data[["Close"]].copy()
    close = close.dropna(how="all")
    return close

def compute_relative_performance(close_df):
    rel = close_df.divide(close_df.iloc[0]) * 100
    return rel

def make_relative_chart(rel_df, chart_path):
    plt.style.use("seaborn-v0_8-whitegrid")

    fig, ax = plt.subplots(figsize=(10, 6))

    for col in rel_df.columns:
        ax.plot(rel_df.index, rel_df[col], label=col)

    ax.set_title("Relative Performance (Last ~2 Months, Close = 100 at Start)")
    ax.set_xlabel("Date")
    ax.set_ylabel("Relative Performance (Start = 100)")
    ax.legend(title="Ticker", loc="best")
```

```
def make_relative_chart(rel_df, chart_path):
    plt.style.use("seaborn-v0_8-whitegrid")

    fig, ax = plt.subplots(figsize=(10, 6))

    for col in rel_df.columns:
        ax.plot(rel_df.index, rel_df[col], label=col)

    ax.set_title("Relative Performance (Last ~2 Months, Close = 100 at Start)")
    ax.set_xlabel("Date")
    ax.set_ylabel("Relative Performance (Start = 100)")
    ax.legend(title="Ticker", loc="best")

    fig.autofmt_xdate()
    fig.tight_layout()
    fig.savefig(chart_path, dpi=150)
    plt.close(fig)

async def main():
    # 1) test text message
    bot = Bot(token=TELEGRAM_BOT_TOKEN)
    await bot.send_message(chat_id=TELEGRAM_CHAT_ID, text="hi this is testing")

    # 2) data and chart
    close = download_data(TICKERS, PERIOD_DAYS)
    rel = compute_relative_performance(close)
    make_relative_chart(rel, CHART_FILENAME)

    # 3) send chart
    caption_chart = "Relative performance: HL vs SLV (last ~2 months, close rebased to 100)."
    with open(CHART_FILENAME, "rb") as f:
        await bot.send_photo(chat_id=TELEGRAM_CHAT_ID, photo=f, caption=caption_chart)

# Jupyter-safe runner
loop = asyncio.get_event_loop()
if loop.is_running():
    task = loop.create_task(main())
else:
    loop.run_until_complete(main())
```