**1. Tạo Bot trên Telegram**

Telegram cung cấp BotFather để tạo bot và lấy token API.

**Bước thực hiện:**

1. Mở Telegram, tìm **BotFather**
2. Nhập lệnh: /newbot
3. Đặt tên cho bot (ví dụ: MyAwesomeBot)
4. Chọn username cho bot (phải kết thúc bằng bot, ví dụ: MyAwesomeBot123\_bot)

Nhận **token API**, dạng như:  
7805318219:AAEF0RIVRO-NziFbksuy3X\_yfZ\_nKUCe9S4

**2. Thiết lập Server để xử lý tin nhắn**

Có thể sử dụng **Node.js, Python, Java, hoặc bất kỳ ngôn ngữ nào hỗ trợ HTTP API**. Dưới đây là 2 cách để nhận tin nhắn từ người dùng:

**Cách 1: Long Polling (Dễ triển khai, phù hợp với dev nhỏ)**

Bot sẽ liên tục gửi request đến Telegram để lấy tin nhắn mới.

* Gửi request getUpdates để lấy tin nhắn mới:  
   https://api.telegram.org/bot<TOKEN>/getUpdates

* Sau khi lấy được tin nhắn, xử lý logic bot.

**Ưu điểm:** Không cần server chạy liên tục.  
**Nhược điểm:** Không phù hợp với bot có lượng người dùng lớn.

**Cách 2: Webhook (Tốt hơn, dùng cho production)**

Telegram sẽ gửi tin nhắn về server của bạn khi có user tương tác.

Bước 1: Đặt Webhook với API Telegram:  
 https://api.telegram.org/bot<TOKEN>/setWebhook?url=<YOUR\_SERVER\_URL>

Bước 2: Xây dựng API để nhận tin nhắn từ Telegram.

**Ưu điểm:** Nhanh hơn, tiết kiệm tài nguyên.  
**Nhược điểm:** Cần có server hoạt động 24/7 (có thể dùng Heroku, AWS, VPS...).

**3. Xây dựng Logic Bot**

Tùy vào mục đích, bạn có thể thiết kế bot trả lời tự động, hỗ trợ khách hàng, chatbot AI (kết hợp GPT API), v.v.

- Xử lý tin nhắn text

- Gửi ảnh, video, file

- Tạo menu phím bấm

- Kết nối với database

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

**4. Triển khai lên Server**

* Nếu dùng Webhook: Host API bằng Flask/Django (Python), Express.js (Node.js), Spring Boot (Java), v.v.
* Nếu dùng Long Polling: Chạy bot trên máy chủ.

**5. Mở rộng & Tối ưu hóa**

* Kết nối với **ChatGPT API** để tạo bot AI thông minh.
* Dùng **Redis** để cache dữ liệu, tăng tốc độ.
* Quản lý tin nhắn với **MySQL/PostgreSQL/MongoDB**.
* Xây dựng giao diện quản trị bot trên Web.

Hiểu luồng hoạt động của một con bot Telegram chạy bằng Python có thể giúp bạn hình dung rõ cách dữ liệu di chuyển từ Telegram đến code của bạn và ngược lại.

**Flow (Luồng hoạt động) của Bot Telegram**

Ảnh có chứa văn bản, Phông chữ, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

**1.Người dùng gửi tin nhắn trên Telegram**

* Khi người dùng nhắn tin cho bot trên Telegram, tin nhắn này được gửi đến máy chủ của Telegram.

**2.Telegram gửi dữ liệu tin nhắn đến bot của bạn**

* Nếu dùng **Long Polling**, bot của bạn liên tục gửi request đến Telegram để "hỏi" xem có tin nhắn mới không.
* Nếu dùng **Webhook**, Telegram tự động gửi tin nhắn đến server của bạn khi có người nhắn tin.

**3.Bot nhận tin nhắn và xử lý logic**

* Bot đọc nội dung tin nhắn, kiểm tra đó là lệnh hay văn bản bình thường.
* Nếu là lệnh (/start, /help), bot gọi các hàm tương ứng.
* Nếu là tin nhắn thông thường, bot có thể phản hồi lại hoặc gọi AI xử lý.

**4.Bot gửi phản hồi lại Telegram**

* Sau khi xử lý, bot gửi phản hồi qua API của Telegram, và Telegram hiển thị tin nhắn đó cho người dùng.

**Tóm tắt đơn giản:**

Người dùng nhắn tin → Telegram nhận tin nhắn → Bot xử lý (chạy Python) → Gửi phản hồi về Telegram → Hiển thị cho người dùng.

**Bot đọc file Python và chạy như thế nào?**

Khi bạn **chạy file Python** có chứa code bot, nó sẽ hoạt động như một chương trình nền (background process).

* Nếu dùng **Long Polling**, file Python chạy liên tục để kiểm tra tin nhắn mới.
* Nếu dùng **Webhook**, file Python tạo một server (ví dụ Flask/Django) để lắng nghe request từ Telegram.

**Ví dụ với Long Polling (chạy trên terminal):**

python my\_telegram\_bot.py

Lệnh này khởi động bot, nó liên tục chạy và lắng nghe tin nhắn mới từ Telegram.

**Nếu có nhiều file Python thì bot hoạt động thế nào?**

Thông thường, chúng ta **tổ chức mã nguồn thành nhiều file** để dễ quản lý, thay vì viết tất cả code trong một file duy nhất.

**1️ Cấu trúc thư mục phổ biến cho bot Telegram**

my\_telegram\_bot/

│── main.py               # File chính chạy bot

│── handlers.py           # Chứa các hàm xử lý tin nhắn

│── config.py             # Chứa token và cài đặt bot

│── database.py           # Kết nối database (nếu có)

│── requirements.txt      # Danh sách thư viện cần cài

**2️ Chạy bot với nhiều file**

Trong main.py, bạn **import** các module từ file khác:

from handlers import start\_handler, text\_handler

from telegram.ext import Updater, CommandHandler, MessageHandler, Filters