**What is a Webhook?**

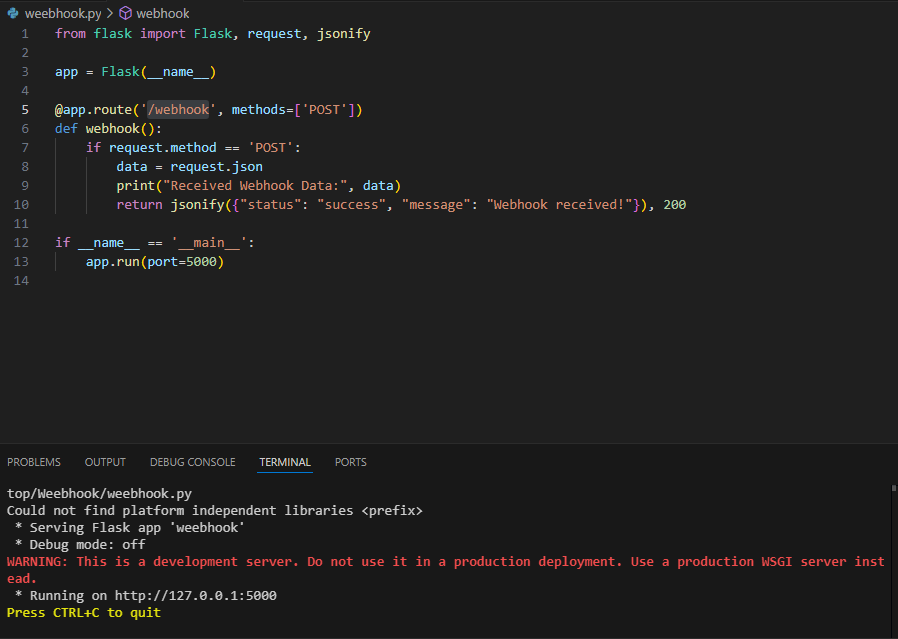
**A webhook is a way for one application (let's say App A) to send real-time data to another application (App B) when a specific event happens. Instead of App B constantly checking (polling) App A for updates, App A pushes the data to App B using an HTTP POST request.**

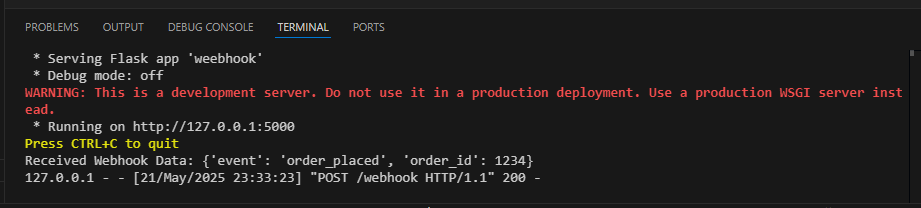
**It's like subscribing to an event — when the event occurs, your server gets notified automatically.**

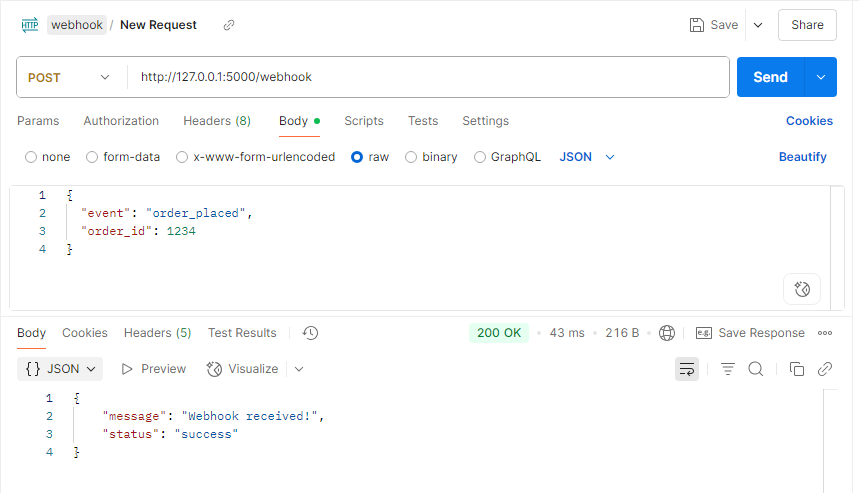
**Real-World Analogy:**

**Imagine ordering pizza. You give the pizza shop your phone number. When the pizza is ready, they call you. You don’t have to keep calling them to ask.**

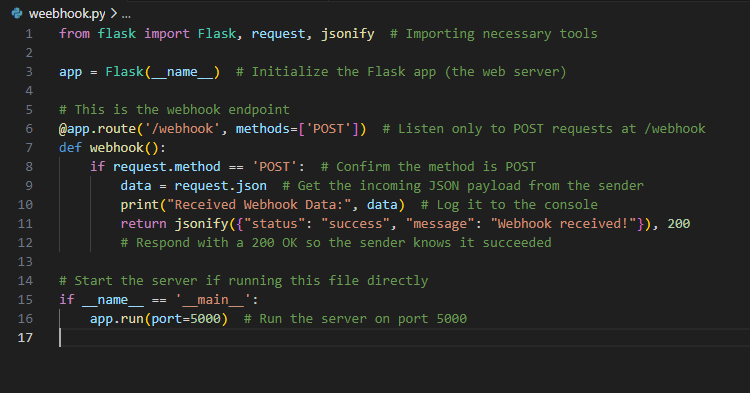
**(Inputs)**





(Output)

**How This Python Program Represents a Webhook**



**Step-by-Step: How This Acts as a Webhook Receiver**

| **Step** | **Action** | **What It Means** |
| --- | --- | --- |
| **1** | **You run this Flask app (python app.py)** | **Starts a local web server** |
| **2** | **The app listens at http://localhost:5000/webhook** | **This is your webhook endpoint** |
| **3** | **Another app (or test tool like Postman or curl) sends a POST request to this URL with JSON data** | **This simulates a webhook sender** |
| **4** | **The webhook() function receives the data** | **Flask processes the POST request** |
| **5** | **It extracts the JSON using request.json** | **You now have the webhook data** |
| **6** | **It prints the data and returns a success message** | **Confirms to the sender that you got the data** |

**Conclusion: Why This is a Webhook**

**This Flask app is a webhook receiver because:**

* **It exposes an HTTP endpoint (/webhook)**
* **It accepts incoming POST requests with event data**
* **It processes the data and returns an acknowledgment**
* **It simulates how real services like Stripe or GitHub notify your system about events**

**❓ Your Question:**

**The Flask webhook example is open to anyone. So if we don't have authentication or subscription, how is this a *real* webhook example?**

**Yes — this is still a valid webhook example.**

**A webhook is not defined by whether users are authenticated or subscribed.  
It is defined by:**

* **✅ The server exposing an HTTP endpoint (like /webhook)**
* **✅ Another system (automated or external) sending an HTTP POST request to that endpoint**
* **✅ The server automatically processing incoming data — no polling required**

**The webhook receiver does not have to allow users to subscribe. It's just waiting for external systems to push data to it when events happen.**

**🔒 Real-World Webhook Security**

**However, in real-world production:**

**Yes, you are absolutely right — we must secure the webhook endpoint.  
Otherwise, anyone can abuse it.**

**🔐 Common Security Approaches:**

| **Method** | **Description** |
| --- | --- |
| **Secret Header Token** | **Sender adds a secret in the Authorization or custom header, receiver validates it** |
| **Signature Verification (HMAC)** | **Sender signs the body using a shared secret, receiver verifies** |
| **IP Whitelisting** | **Accept only requests from trusted IP addresses** |
| **OAuth/Bearer Token** | **Authenticated token in request headers** |
| **Request Rate Limiting** | **Prevent flooding attacks** |

**What is a WebSocket?**

**Concept**

* **A WebSocket is a communication protocol that allows persistent, bidirectional communication between client (like browser) and server over a single connection.**
* **Unlike HTTP (request → response), WebSocket allows:**
  + **📤 Client → Server messages anytime**
  + **📥 Server → Client messages anytime**

**Think of it like a phone call (continuous talk) vs. emails (one request, one reply).**

**✅ Use Cases**

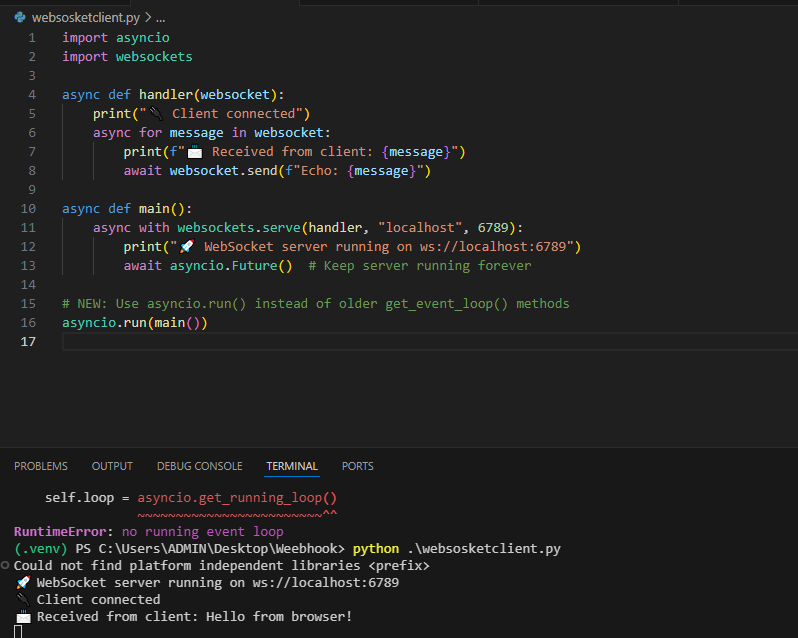
* **💬 Chat apps (real-time messaging)**
* **📈 Live dashboards (stock prices, weather, IoT sensors)**
* **🎮 Multiplayer games**
* **📡 Live notifications**

**⚙️ How It Works**

1. **Client (browser) sends a WebSocket handshake (via HTTP).**
2. **Server responds and upgrades the connection to WebSocket.**
3. **Both sides can now send and receive messages at any time.**

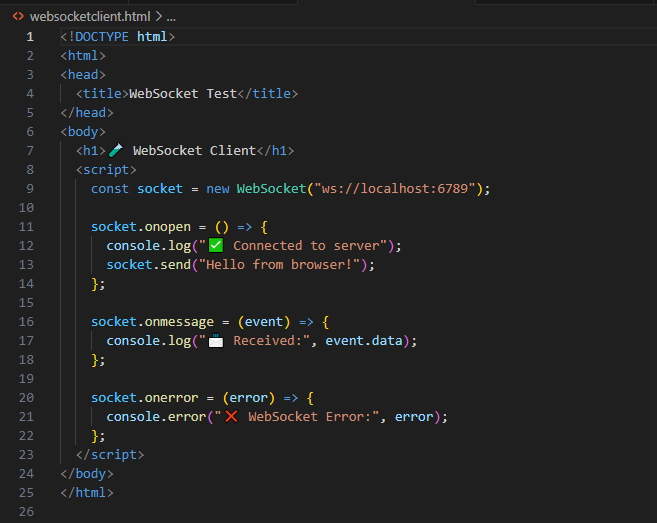
**Program**

**(Server)**

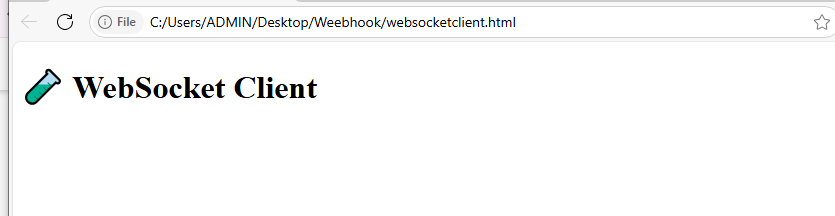


**Client**

**(Code)**



**(Output)**



**🧠 Key Differences: WebSocket vs Webhook**

| **Feature** | **WebSocket** | **Webhook** |
| --- | --- | --- |
| **Connection** | **Persistent (long-lived)** | **One-time HTTP request** |
| **Direction** | **Two-way (client ↔ server)** | **One-way (sender → receiver)** |
| **Use Case** | **Real-time chat, games, dashboard** | **Event notifications (e.g. payment)** |
| **Server Push** | **✅ Yes** | **✅ Yes (but limited, one-time)** |
| **State** | **Maintains connection state** | **Stateless** |