Aquí tienes una **plantilla básica en JSON para Postman** que te permite probar localmente el webhook de WhatsApp (/webhook) de SentinelCore como si Twilio estuviera enviando un mensaje. Puedes importarla directamente en Postman.

**✅ postman\_whatsapp\_webhook\_test.json**

{

"info": {

"name": "Webhook WhatsApp - SentinelCore",

"\_postman\_id": "a9a3fbe7-9d8f-4e7f-9f03-8efbe96d1bfa",

"description": "Simula una llamada de Twilio al webhook /webhook de la API WhatsApp para SentinelCore.",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"

},

"item": [

{

"name": "Simular mensaje WhatsApp",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/x-www-form-urlencoded"

}

],

"body": {

"mode": "urlencoded",

"urlencoded": [

{

"key": "Body",

"value": "status",

"description": "Comando recibido por WhatsApp"

},

{

"key": "From",

"value": "whatsapp:+34600111222",

"description": "Número simulado de origen"

}

]

},

"url": {

"raw": "http://localhost:5005/webhook",

"protocol": "http",

"host": ["localhost"],

"port": "5005",

"path": ["webhook"]

}

},

"response": []

}

]

}

**📦 ¿Qué hacer con esto?**

1. Abre **Postman**.
2. Menú → Importar.
3. Elige la pestaña **Raw text** y pega el contenido JSON.
4. Ejecuta la petición POST con el mensaje deseado (status, ayuda, escanear, etc.).
5. Revisa la consola de Flask (puerto 5005) y el registro de logs.

¿Quieres que te lo empaquete como archivo .json descargable directamente o continúo con la parte de autenticación (autenticacion.py con JWT)?