

OpenAI Platform

Realtime API with SIP

[Copy page](#)

Connect to the Realtime API using SIP.

SIP is a protocol used to make phone calls over the internet. With SIP and the Realtime API you can direct incoming phone calls to the API.

Overview

If you want to connect a phone number to the Realtime API, use a SIP trunking provider (e.g., Twilio). This is a service that converts your phone call to IP traffic. After you purchase a phone number from your SIP trunking provider, follow the instructions below.

Start by creating a webhook for incoming calls, at platform.openai.com. Then, point your SIP trunk at the OpenAI SIP endpoint, using the project ID for which you configured the webhook, e.g.,

```
sip:$PROJECT_ID@sip.api.openai.com;transport=UDP
```

To find your `$PROJECT_ID`, go to your [settings] > **General**. The page displays the project ID. It should have a `proj_` prefix.

When OpenAI receives SIP traffic associated with your project, the webhook that you configured will

Overview

[Connection details](#)[Handling the Webhook](#)[Redirect the call](#)

be fired. The event fired will be a `realtime.call.incoming` event.

This webhook lets you accept or reject the call. When accepting the call, you'll provide the configuration (instructions, voice, etc) for the Realtime API session. Once established, you can set up a web socket and monitor the session as usual. The APIs to accept, reject, and monitor the call are documented below.

Connection details

URIs used for interacting with Realtime API and SIP:

SIP URI	<code>sip:\$PROJECT_ID@sip.api.openai.com;trans</code>
----------------	--

Accept URI	<code>https://api.openai.com/v1/realtime/calls</code>
-------------------	---

Reject URI	<code>https://api.openai.com/v1/realtime/calls</code>
-------------------	---

Refer URI	<code>https://api.openai.com/v1/realtime/calls</code>
------------------	---

Events URI	<code>wss://api.openai.com/v1/realtime?call_id</code>
-------------------	---

Find your `$CALL_ID` in the `call_id` field in data object present in the webhook. See an example in the next section.

Handle the webhook

The following is an example of a

`realtime.call.incoming` handler. It accepts the call and then logs all the events from the Realtime API.

Python

Python

python ↕



```
1  from flask import Flask, request, Response
2  from openai import OpenAI, InvalidWebhook
3  import asyncio
4  import json
5  import os
6  import requests
7  import time
8  import threading
9  import websockets
10
11 app = Flask(__name__)
12 client = OpenAI(
13     webhook_secret=os.environ["OPENAI_W
14 )
15
16 AUTH_HEADER = {
17     "Authorization": "Bearer " + os.get
18 }
19
20 call_accept = {
21     "type": "realtime",
22     "instructions": "You are a support
23     "model": "gpt-4o-realtime-preview-2
24 }
25
26 response_create = {
27     "type": "response.create",
28     "response": {
29         "instructions": (
30             "Say to the user 'Thank you
31         )
32     },
```

```
33 }
34
35
36 async def websocket_task(call_id):
37     try:
38         async with websockets.connect(
39             "wss://api.openai.com/v1/re
40             additional_headers=AUTH_HEA
41         ) as websocket:
42             await websocket.send(json.c
43
44             while True:
45                 response = await websoc
46                 print(f"Received from W
47     except Exception as e:
48         print(f"WebSocket error: {e}")
49
50
51 @app.route("/", methods=["POST"])
52 def webhook():
53     try:
54         event = client.webhooks.unwrap(
55
56         if event.type == "realtime.call
57             requests.post(
58                 "https://api.openai.com
59                 + event.data.call_id
60                 + "/accept",
61                 headers={**AUTH_HEADER,
62                 json=call_accept,
63             )
64             threading.Thread(
65                 target=lambda: asyncio.
66                 websocket_task(event
67             ),
68                 daemon=True,
69             ).start()
70             return Response(status=200)
71     except InvalidWebhookSignatureError:
72         print("Invalid signature", e)
73         return Response("Invalid signat
74
```

```
75
76 if __name__ == "__main__":
77     app.run(port=8000)
```

Redirect the call

It's also possible to redirect the call to another number. During the call, make a POST to the `refer` endpoint:

URL `https://api.openai.com/v1/realtime/cal`

Payload JSON with one key `target_uri`

This is the value used in the Refer-To. You can use `tel:+14152909007`

Headers `Authorization: Bearer YOUR_API_KEY`

Substitute `YOUR_API_KEY` with a [standard API](#)

Next steps

Now that you've connected over SIP, use the left navigation or click into these pages to start building your realtime application.

[Using realtime models](#)

[Managing conversations](#)

[Webhooks and server-side controls](#)

[Realtime transcription](#)

