

# OpenAI Platform

## Realtime API

 Copy page

Build low-latency, multimodal LLM applications with the Realtime API.

The OpenAI Realtime API enables low-latency communication with models that natively support speech-to-speech interactions as well as multimodal inputs (audio, images, and text) and outputs (audio and text). These APIs can also be used for realtime audio transcription.

## Voice agents

One of the most common use cases for the Realtime API is building voice agents for speech-to-speech model interactions in the browser. Our recommended starting point for these types of applications is the Agents SDK for TypeScript, which uses a WebRTC connection to the Realtime model in the browser, and WebSocket when used on the server.

### Overview

Voice agents

Connection methods

API usage

Build voice agents 

Build voice agents in the browser.

```
1 import { RealtimeAgent, RealtimeSession } from 'openai';
2
3 const agent = new RealtimeAgent({
4   name: "Assistant",
5   instructions: "You are a helpful assistant.",
6 });
7
8 const session = new RealtimeSession(agent);
9
10 // Automatically connects your microphone
11 await session.connect({
12   apiKey: "<client-api-key>",
13 });
```



### Voice Agent Quickstart

Follow the voice agent quickstart to build Realtime agents in the browser.

To use the Realtime API directly outside the context of voice agents, check out the other connection options below.

## Connection methods

While building voice agents with the Agents SDK is the fastest path to one specific type of application, the Realtime API provides an entire suite of flexible tools for a variety of use cases.

There are three primary supported interfaces for the Realtime API:



### WebRTC connection

Ideal for browser and client-side interactions with a Realtime model.



### WebSocket connection

Ideal for middle tier server-side applications with consistent low-latency network connections.



### SIP connection

Ideal for VoIP telephony connections.

Depending on how you'd like to connect to a Realtime model, check out one of the connection guides above to get started. You'll learn how to initialize a Realtime session, and how to interact with a Realtime model using client and server events.

## API Usage

Once connected to a realtime model using one of the methods above, learn how to interact with the model in these usage guides.

**Prompting guide:** learn tips and best practices for prompting and steering Realtime models.

**Inputs and outputs:** Learn how to pass audio, text, and image inputs to the model, and how to receive audio and text back.

**Managing conversations:** Learn about the Realtime session lifecycle and the key events that happen during a conversation.

**Webhooks and server-side controls:** Learn how you can control a Realtime session on the server to call tools and implement guardrails.

**Function calling:** Give the realtime model access to call custom code in your own application when appropriate.

**MCP servers:** Give realtime models access to new capabilities via Model Context Protocol (MCP) servers.

**Realtime audio transcription:** Transcribe audio streams in real time over a WebSocket connection.