

## Get Started

Introduction

Get Started With Device

Update OMI Firmware

## Developer

Apps ▾

Introduction

Prompt-Based Apps

Developing Integration Apps for  
OMI

Real-Time Audio Streaming

Import Apps

Submitting Apps

OMI OAuth 2.0 Integration

Notifications

App Setup

Backend ▸

Firmware ▸

SDK ▸

App-device protocol

Contribution

APIs &amp; Integrations ▸

Audio &amp; Testing ▸

## Hardware

Omi

Omi DevKit 2

Omi DevKit 1

omiGlass DevKit

## DIY Guide

Build Your Own OMI Device

Part List

Building the Device

## Info

Support

License

Disclaimer

Privacy Policy

## Apps

## Real-Time Audio Streaming

## Streaming Real-Time Audio From Device to Anywhere

Omi allows you to stream audio bytes from your DevKit1 or DevKit2 directly to your backend or any other service, enabling you to perform various analyses and store the data. You can also define how frequently you want to receive the audio bytes.

## Step 1: Create an Endpoint

Create an endpoint (webhook) that can receive and process the data sent by our backend. Our backend will make a POST request to your webhook with `sample_rate` and `uid` as query parameters. The request from our backend will look like this:

```
POST /your-endpoint?sample_rate=16000&uid=user123
```

The data sent is of type `octet-stream`, which is essentially a stream of bytes. You can either create your own endpoint or use the example provided below. Once it's ready and deployed, proceed to the next step.

Note: The sample rate refers to the audio samples captured per second. DevKit1 (v1.0.4 and above) and DevKit2 record audio at a sample rate of 16,000 Hz, while DevKit1 with v1.0.2 records at a sample rate of 8,000 Hz. The `uid` represents the unique ID assigned to the user in our system.

## Step 2: Add the Endpoint to Developer Settings

1. Open the Omi App on your device.
2. Go to Settings and click on Developer Mode.
3. Scroll down until you see Realtime audio bytes, and set your endpoint there.
4. In the Every x seconds field, define how frequently you want to receive the bytes. For example, if you set it to 10 seconds, you will receive the audio bytes every 10 seconds.

That's it! You should now see audio bytes arriving at your webhook. The audio bytes are raw, but you can save them as audio files by adding a WAV header to the accumulated bytes.

Check out the example below to see how you can save the audio bytes as audio files in Google Cloud Storage using the audio streaming feature.

## Example: Saving Audio Bytes as Audio Files in Google Cloud Storage

Step 1: Create a Google Cloud Storage bucket and set the appropriate permissions. You can follow the steps mentioned [here](#) up to step 5.

Step 2: Fork the example repository from [github.com/mdmohsin7/omi-audio-streaming](https://github.com/mdmohsin7/omi-audio-streaming).

Step 3: Clone the repository to your local machine.

Step 4: Deploy it to any of your preferred cloud providers like GCP, AWS, DigitalOcean, or run it locally (you can use Ngrok for local testing). The repository includes a Dockerfile for easy deployment.

Step 5: While deploying, ensure the following environment variables are set:

- `GOOGLE_APPLICATION_CREDENTIALS_JSON` : Your GCP credentials, encoded in base64.
- `GCS_BUCKET_NAME` : The name of your GCP storage bucket.

Step 6: Once the deployment is complete, set the endpoint in the Developer Settings of the Omi App under

Realtime audio bytes. The endpoint should be the URL where you deployed the example + `/audio` .

Step 7: You should now see audio files being saved in your GCP bucket every x seconds, where x is the value you set in the `Every x seconds` field.

## Contributing 🙌

We welcome contributions from the open source community! Whether it's improving documentation, adding new features, or reporting bugs, your input is valuable. Check out our [Contribution Guide](#) for more information.

## Support

If you're stuck, have questions, or just want to chat about Omi:

- **GitHub Issues:** 🐛 For bug reports and feature requests
- **Community Forum:** 💬 Join our [community forum](#) for discussions and questions
- **Documentation:** 📖 Check out our [full documentation](#) for in-depth guides

Happy coding! 🖥️ If you have any questions or need further assistance, don't hesitate to reach out to our community.

< [Developing Integration Apps for OMI](#)

[Import Apps](#) >



Powered by Mintlify