LiveKit SIP Outbound Integration Guide

TxContact SIP Integration Setup

---

1. Overview

We are integrating LiveKit with TxContact SIP server for outbound AI calling.

Provided by TxContact:

- SIP Server IP: 192.168.0.48\
- SIP Port: 5060\
- RTP Port Range: 16384 to 32768\
- Authentication: IP based\
- Our IP is already whitelisted

This document explains exactly what needs to be configured on the telephony and LiveKit side.

---

2. Network Prerequisites

2.1 Verify Network Connectivity

From LiveKit server:

```
ping 192.168.0.48
nc -vu 192.168.0.48 5060
```

SIP server must be reachable over internal network.

If not reachable, fix routing or firewall.

2.2 Firewall Configuration

On LiveKit server allow:

Outbound: - UDP 5060\

- UDP 16384 to 32768

Inbound: - UDP 16384 to 32768

If RTP ports are blocked, call will connect but audio will fail.

3. LiveKit Installation

Ensure LiveKit is installed and running inside same internal network.

LiveKit config must not use external IP since internal network is used.

In `livekit.yaml`:

```
rtc:
  use_external_ip: false
```

Restart LiveKit after configuration.

4. Create Outbound SIP Trunk

Create outbound SIP trunk in LiveKit pointing to TxContact SIP server.

Required Configuration

- SIP Address: 192.168.0.48\
- Port: 5060\
- Transport: UDP\
- Authentication: None\
- Caller ID: Approved CLI number

Since authentication is IP based, do not configure username or password.

Example using CLI:

```
lk sip outbound create   --name txcontact-trunk   --address 192.168.0.48   --port 5060   -
```

After creation, share trunk ID with backend team.

5. Codec Configuration

5. Codec Configuration

Provider is expected to support G711 A law.

Ensure LiveKit prioritizes:

- PCMA G711 A law

If codec mismatch occurs, call may connect but audio will not work.

6. Outbound Call Flow

When backend triggers call:

1. LiveKit creates SIP participant using trunk ID\
2. SIP INVITE sent to 192.168.0.48\
3. TxContact routes call to PSTN\
4. RTP media flows between LiveKit and SIP server\
5. AI agent joins same LiveKit room\
6. Audio bridged automatically

Telephony side must ensure SIP signaling and RTP flow are stable.

7. Testing Procedure

Before production:

Step 1

Confirm:\

- Caller ID\
- Dial format\
- CPS limit\
- Concurrent channel limit

Step 2

Place controlled outbound test call.

Verify:

- Call rings\

- Call connects\
- Two way audio works\
- Call disconnect works

Step 3

Check logs if failure:

- SIP response codes\
- RTP packet flow\
- Codec negotiation

---

8. CPS and Channel Handling

Confirm provider limits.

If example limit is:

- 2 CPS\
- 20 concurrent channels

Ensure LiveKit or backend rate limits outbound call attempts accordingly.

Do not exceed provider limits.

---

9. Responsibilities

Telephony Setup Owner

- LiveKit installation\
- SIP trunk creation\
- Firewall configuration\
- RTP configuration\
- Codec alignment\
- Basic call testing\
- Telephony level debugging

Backend Team

- Call triggering via LiveKit API\
- AI agent joining room\

- STT LLM TTS pipeline\
- Conversation logic

10. Common Failure Scenarios

Call connects but no audio

Cause: RTP ports blocked or codec mismatch

403 Forbidden

Cause: Incorrect Caller ID

404 Not Found

Cause: Wrong dial format

488 Not Acceptable

Cause: Codec mismatch

11. Final Deliverables

Before handing over:

- LiveKit running and stable\
- Outbound SIP trunk created\
- Trunk ID shared\
- Test call successful with two way audio\
- CPS and channel limits confirmed

Once this is completed, backend team will integrate AI call orchestration