

FreePBX 17.0.21 : ARI を使用して着信を Node.js 側へ通知する構成手順

■ 概要

本ドキュメントでは、FreePBX 17.0.21（Asterisk 20.x 系）で **Asterisk REST Interface（ARI）** を有効化し、着信イベントを Node.js アプリに通知する仕組みを構築する手順をまとめます。

🌀 構成イメージ



⚙️ 環境要件

項目	推奨バージョン
FreePBX	17.0.21
Asterisk	20.x
Node.js	18 以上（LTS推奨）
OS	Rocky Linux / Debian / Ubuntu (WSL可)

① Asterisk 側 : ARIを有効化

1.1 /etc/asterisk/http.conf を編集

```
[general]
enabled=yes
bindaddr=0.0.0.0
bindport=8088
```

設定説明:

- ◇ `bindaddr=0.0.0.0` : どのネットワークインターフェースからも接続を受ける
- ◇ `bindport=8088` : ARI/HTTP用ポート（必要に応じて変更可）

1.2 FreePBX GUI で ARI ユーザーを追加

1. 管理画面にログイン
2. メニュー : **Admin** → **Asterisk REST Interface Users**
3. 「**Add User**」をクリック
4. 次の内容を入力 :

設定項目	値
Username	ariuser
Password	任意（例 : YOUR_ARIPASS）
Read Only	No
Read/Write Permissions	All

5. 「**Submit**」 → 「**Apply Config**」

1.3 設定反映

```
sudo fwconsole restart
```

ブラウザで確認 :

```
http://<asterisk_ip>:8088/ari/api-docs/
```

ARIのSwagger UIが表示されれば有効化成功です。

2 FreePBX : 着信を一度 ARI に通す

FreePBX のGUIだけでは `Stasis()` アプリを呼び出す設定はできません。そのため、`extensions_custom.conf` にコンテキストを追加し、GUI上で Custom Destination を作成してルーティングします。

2.1 `/etc/asterisk/extensions_custom.conf` を編集

```
[incoming-ari]
exten => s,1,NoOp(=== incoming via ARI entry ===)
same => n,Stasis(myapp,${CALLERID(num)},${EXTEN})
same => n,Hangup()
```

注意: ここで **myapp** は後述の Node.js 側 ARI アプリ名です。

2.2 FreePBX GUI : Custom Destinationの作成

1. **Admin** → **Custom Destinations**
2. **「Add Custom Destination」**
3. **以下を設定 :**

項目	値
Target	incoming-ari,s,1
Description	To ARI

4. **「Submit」** → **「Apply Config」**

2.3 Inbound Route に紐付け

1. **Connectivity** → **Inbound Routes**
2. **対象の DID を選択**
3. **Set Destination** → **「Custom Destination → To ARI」**
4. **「Submit」** → **「Apply Config」**

補足

このドキュメントは FreePBX での ARI 設定の基本手順を示しています。Node.js 側の実装については、別途 **asterisk-backend** のドキュメントを参照してください。

関連ドキュメント:

- [ASTERISK_SETUP.md](#) - 詳細なAsterisk設定手順
- [asterisk-backend/README.md](#) - Node.js側の実装詳細