

# Channel Presets Troubleshooting Guide

---

## Overview

---

This guide helps you diagnose and fix issues with the Channel Presets feature, particularly the “Failed to fetch channel presets” error.







## Quick Fix

---

If you’re experiencing the “Failed to fetch channel presets” error, run the automated fix script:

```
cd /home/ubuntu/Sports-Bar-TV-Controller
./scripts/fix-channel-presets.sh
```

This script will:

1.  Backup your database
2.  Verify/create database structure
3.  Apply all necessary migrations
4.  Regenerate Prisma client
5.  Restart the server
6.  Verify the fix worked

## Diagnostic Tool

---

To diagnose issues without making changes, run:

```
./scripts/diagnose-channel-presets.sh
```

This will check:

- Database file existence and location
- Database schema and table structure
- Prisma client generation status
- Migration status
- Server status
- API endpoint functionality

## Common Issues and Solutions

---

### Issue 1: “Failed to fetch channel presets” Error

#### Symptoms:

- Error message in Channel Presets tab
- “Error loading presets” notification
- Empty presets list

#### Causes:

1. Database migrations not applied

2. Prisma client not generated
3. Server not restarted after updates
4. Database file missing or corrupted

**Solution:**

```
./scripts/fix-channel-presets.sh
```

Or manually:

```
# 1. Generate Prisma client
npx prisma generate

# 2. Apply migrations
npx prisma migrate deploy

# 3. Restart server
pm2 restart sports-bar-tv-controller

# 4. Verify
curl http://localhost:3000/api/channel-presets
```

## Issue 2: Database Not Found

**Symptoms:**

- “Database file not found” in diagnostic
- Prisma errors about missing database

**Solution:**

The fix script will automatically find or create the database in the correct location. If you need to do it manually:

```
# Check .env for database path
cat .env | grep DATABASE_URL

# Create database directory if needed
mkdir -p prisma/data

# Run migrations to create database
npx prisma migrate deploy
```

## Issue 3: Missing ChannelPreset Table

**Symptoms:**

- “ChannelPreset table does not exist” error
- API returns database errors

**Solution:**

```
# Apply migrations
npx prisma migrate deploy

# If that fails, try db push
npx prisma db push
```

## Issue 4: Prisma Client Out of Date

### Symptoms:

- "ChannelPreset model not found in client"
- TypeScript errors about missing types

### Solution:

```
# Regenerate Prisma client
npx prisma generate

# Restart server
pm2 restart sports-bar-tv-controller
```

## Issue 5: Server Not Responding

### Symptoms:

- API endpoint returns connection errors
- Port 3000 not in use

### Solution:

```
# Check PM2 status
pm2 status

# Restart server
pm2 restart sports-bar-tv-controller

# If not in PM2, start it
pm2 start npm --name sports-bar-tv-controller -- start
pm2 save
```

## Manual Verification Steps

---

### 1. Check Database Structure

```
# Connect to database
sqlite3 prisma/dev.db

# List tables
.tables

# Check ChannelPreset structure
PRAGMA table_info(ChannelPreset);

# Count records
SELECT COUNT(*) FROM ChannelPreset;

# Exit
.quit
```

Expected columns:

- id (TEXT, PRIMARY KEY)
- name (TEXT)
- channelNumber (TEXT)

- `deviceType` (TEXT)
- `order` (INTEGER)
- `isActive` (BOOLEAN)
- `usageCount` (INTEGER)
- `lastUsed` (DATETIME, nullable)
- `createdAt` (DATETIME)
- `updatedAt` (DATETIME)

## 2. Test API Endpoint

```
# Test GET endpoint
curl http://localhost:3000/api/channel-presets

# Expected response:
# {"success":true,"presets":[]}

# Test with pretty printing
curl -s http://localhost:3000/api/channel-presets | jq '.'
```

## 3. Check Server Logs

```
# View PM2 logs
pm2 logs sports-bar-tv-controller

# View last 50 lines
pm2 logs sports-bar-tv-controller --lines 50

# Follow logs in real-time
pm2 logs sports-bar-tv-controller --lines 0
```

## 4. Check Browser Console

1. Open browser DevTools (F12)
2. Go to Console tab
3. Look for errors related to:
  - `/api/channel-presets`
  - Network errors
  - JavaScript errors

## Database Migration Details

---

The Channel Presets feature requires two migrations:

### Migration 1: Create ChannelPreset Table

**File:** `prisma/migrations/20250103_channel_presets/migration.sql`

Creates the base table with:

- Basic fields (id, name, channelNumber, deviceType)
- Display order and active status
- Timestamps
- Indexes for performance

## Migration 2: Add Usage Tracking

**File:** `prisma/migrations/20250103_add_usage_tracking/migration.sql`

Adds:

- `usageCount` field (for AI-powered auto-reordering)
- `lastUsed` field (tracks last usage time)
- Index on `usageCount` for efficient sorting

## After Running Update Script

If you've just run `./update_from_github.sh`, the Channel Presets feature should work automatically. However, if you merged PRs #47, #48, #49 manually without running the update script, you need to:

### 1. Pull latest changes:

```
bash
git pull origin main
```

### 2. Run the update script:

```
bash
./update_from_github.sh
```

This will:

- Apply database migrations
- Generate Prisma client
- Restart the server

### 1. Or run the fix script directly:

```
bash
./scripts/fix-channel-presets.sh
```

## Backup and Recovery

### Automatic Backups

The fix script automatically backs up your database before making changes:

- Location: `~/sports-bar-backups/database-backups/`
- Format: `pre-fix-backup-YYYYMMDD-HHMMSS.db`

### Manual Backup

```
# Create backup directory
mkdir -p ~/sports-bar-backups/database-backups

# Backup database
cp prisma/dev.db ~/sports-bar-backups/database-backups/manual-backup-$(date +%Y%m%d-%H%M%S).db
```

## Restore from Backup

```
# Stop server
pm2 stop sports-bar-tv-controller

# Restore database
cp ~/sports-bar-backups/database-backups/backup-file.db prisma/dev.db

# Restart server
pm2 restart sports-bar-tv-controller
```

## Prevention

---

To prevent this issue in the future:

1. **Always run the update script after pulling changes:**

```
bash
./update_from_github.sh
```

2. **The update script handles:**

- Database migrations
- Prisma client generation
- Server restart
- Dependency updates

3. **Don't skip the update script** - it ensures all components are synchronized

## Getting Help

---

If you're still experiencing issues after trying these solutions:

1. **Run the diagnostic script:**

```
bash
./scripts/diagnose-channel-presets.sh
```

2. **Check the logs:**

```
```bash
# Fix script log
cat channel-presets-fix.log
```

# Server logs

```
pm2 logs sports-bar-tv-controller -lines 100
```
```

1. **Provide this information when asking for help:**

- Output of diagnostic script
- Relevant log entries
- Browser console errors
- Steps you've already tried

## Technical Details

### Database Schema

```
model ChannelPreset {
  id          String    @id @default(cuid())
  name        String
  channelNumber String
  deviceType   String
  order        Int       @default(0)
  isActive     Boolean   @default(true)
  usageCount   Int       @default(0)
  lastUsed     DateTime?
  createdAt    DateTime  @default(now())
  updatedAt    DateTime  @updatedAt

  @@index([deviceType, order])
  @@index([isActive])
  @@index([usageCount])
}
```

### API Endpoints

- **GET** /api/channel-presets - List all presets
- **POST** /api/channel-presets - Create preset
- **GET** /api/channel-presets/[id] - Get specific preset
- **PUT** /api/channel-presets/[id] - Update preset
- **DELETE** /api/channel-presets/[id] - Delete preset
- **POST** /api/channel-presets/reorder - Reorder presets
- **POST** /api/channel-presets/tune - Tune to preset
- **GET** /api/channel-presets/statistics - Get usage statistics

### Required Environment Variables

The Channel Presets feature doesn't require any special environment variables. It uses the standard database configuration:

```
DATABASE_URL="file:./prisma/dev.db"
```

## Summary

The Channel Presets feature is a powerful tool for quick channel access. If you encounter the "Failed to fetch channel presets" error:

1. ✓ Run `./scripts/fix-channel-presets.sh` for automated fix
2. ✓ Or run `./scripts/diagnose-channel-presets.sh` to identify the issue
3. ✓ Always use `./update_from_github.sh` after pulling changes
4. ✓ Check logs and browser console for additional clues

The fix script is designed to be safe and will backup your database before making any changes.