Sports Bar TV Controller - Deployment Fix Instructions

Date: October 19, 2025
Issue: 500 errors, input gains showing mock data, zone controls not functioning

---

🔍 Problem Analysis

After comprehensive code analysis, **the application code is correct and properly implemented**. The issues are related to **deployment configuration**, not code bugs.

✅ What's Working in the Code:

- Atlas TCP communication on port 5321 ✓
- JSON-RPC 2.0 protocol implementation ✓
- Proper 0-based/1-based indexing conversion ✓
- Input gain API with real hardware queries ✓
- Zone control API with real-time status ✓
- Error handling and timeouts ✓

❌ What's Likely Broken on the Server:

1. Database not initialized or corrupted
2. No audio processor configured in database
3. Multiple application instances running
4. Network connectivity issues to Atlas processor

🚀 Quick Fix (Automated)

Step 1: Access Your Server

Connect to your server at **24.123.187.42** via RDP or SSH.

Step 2: Navigate to Project Directory

```
cd /path/to/Sports-Bar-TV-Controller
# Usually: cd ~/Sports-Bar-TV-Controller or cd /opt/Sports-Bar-TV-Controller
```

Step 3: Run the Fix Script

```
./fix_deployment.sh
```

This script will automatically:
- Stop conflicting processes
- Configure environment variables

- Initialize database
- Generate Prisma client
- Add Atlas processor configuration
- Build and start the application

Step 4: Test the Application

```
./test_atlas_integration.sh
```

This will verify:
- Application is running
- Atlas processor is reachable
- Input gains are pulling real data
- Zone controls are working

🔧 Manual Fix (If Automated Script Fails)

Fix 1: Stop All Running Processes

```
# Stop PM2 processes
pm2 stop all
pm2 delete all

# Kill any remaining Node.js processes
pkill -f "node.*next"
```

Fix 2: Configure Environment

```
# Create .env file if it doesn't exist
cat > .env << 'EOF'
DATABASE_URL="file:./prisma/dev.db"
NODE_ENV="production"
NEXTAUTH_URL="http://localhost:3000"
NEXTAUTH_SECRET="your-random-secret-here"
EOF
```

Fix 3: Initialize Database

```
# Install dependencies
npm install

# Generate Prisma client
npx prisma generate

# Push database schema
npx prisma db push
```

Fix 4: Add Atlas Processor to Database

```
# Using SQLite directly
sqlite3 prisma/dev.db << 'SQL'
INSERT INTO AudioProcessor (id, name, model, ipAddress, port, tcpPort, zones, status,
createdAt, updatedAt)
VALUES (
    'atlas-main-001',
    'Main Atlas Processor',
    'AZMP8',
    '192.168.1.100',
    80,
    5321,
    8,
    'offline',
    datetime('now'),
    datetime('now')
);
SQL
```

Or using the API (after starting the app):

```
curl -X POST http://localhost:3000/api/audio-processor \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Main Atlas Processor",
    "model": "AZMP8",
    "ipAddress": "192.168.1.100",
    "port": 80,
    "zones": 8,
    "description": "Atlas AZMP8 Audio Processor"
  }'
```

Fix 5: Build and Start Application

```
# Build the application
npm run build

# Start with PM2 (recommended)
pm2 start ecosystem.config.js
pm2 save

# OR start in foreground
npm start
```

🧪 Testing

Test 1: Check Application Status

```
curl http://localhost:3000
# Should return HTML (status 200)
```

Test 2: Check Audio Processors

```
curl http://localhost:3000/api/audio-processor
# Should return JSON with processor list
```

Test 3: Check Input Gains

```
# Replace {processor-id} with actual ID from previous test
curl http://localhost:3000/api/audio-processor/{processor-id}/input-gain
# Should return gain settings from Atlas
```

Test 4: Check Zone Status

```
# Replace {processor-id} with actual ID
curl http://localhost:3000/api/audio-processor/{processor-id}/zones-status
# Should return zone configuration from Atlas
```

Test 5: Test Atlas Connectivity

```
# Test TCP port 5321
nc -zv 192.168.1.100 5321
# Should show "succeeded" or "open"

# Test HTTP port 80
curl -I http://192.168.1.100
# Should return HTTP response
```

🐛 Troubleshooting

Issue: “Database connection error”

Cause: Database not initialized or .env file missing

Solution:

```
# Check if .env exists
cat .env

# Check if database exists
ls -la prisma/dev.db

# Regenerate database
npx prisma db push
```

Issue: “Audio processor not found” (404)

Cause: No processor configured in database

Solution:

```
# Check processors in database
sqlite3 prisma/dev.db "SELECT * FROM AudioProcessor;"

# Add processor (see Fix 4 above)
```

Issue: “Cannot reach Atlas processor”

Cause: Network connectivity issue

Solution:

```
# Test connectivity
ping 192.168.1.100
nc -zv 192.168.1.100 5321

# Check if Atlas is on different IP
# Update processor IP in database:
sqlite3 prisma/dev.db "UPDATE AudioProcessor SET ipAddress='NEW_IP' WHERE id='atlas-main-001';"
```

Issue: “Port 3000 already in use”

Cause: Another instance is running

Solution:

```
# Find process using port 3000
lsof -i :3000

# Kill the process
kill -9 <PID>

# Or use PM2
pm2 stop all
pm2 delete all
```

Issue: Input gains still showing mock data

Cause: Atlas processor not reachable or wrong IP

Solution:
1. Verify Atlas IP address is correct (192.168.1.100)
2. Test connectivity: `nc -zv 192.168.1.100 5321`
3. Check Atlas processor is powered on
4. Verify network routing/firewall rules
5. Check application logs: `pm2 logs`

📊 Verification Checklist

After applying fixes, verify:

- [ ] Application starts without errors
- [ ] Can access http://localhost:3000

- [ ] Audio processor appears in UI
- [ ] Input gain sliders show real values (not -40dB mock data)
- [ ] Zone controls show actual zone names from Atlas
- [ ] Can adjust input gains and see changes
- [ ] Can control zones (volume, mute, source)
- [ ] No 500 errors in browser console
- [ ] No database errors in logs

📝 Important Notes

About the Code

- **No code changes are required** - the implementation is correct
- All Atlas communication uses proper TCP port 5321
- JSON-RPC 2.0 protocol is correctly implemented
- Indexing conversion (0-based ↔ 1-based) is handled properly

About Mock Data

- The code does NOT use mock data for Atlas communication
- If you see mock data, it means:
- Atlas processor is not reachable
- No processor is configured in database
- Database connection failed

About the 500 Errors

- 500 errors are caused by:
- Database connection failures
- Missing processor configuration
- Uncaught exceptions in API routes
- Check logs for specific error messages

🆘 Getting Help

If issues persist after following these instructions:

1. **Check Logs:**
   ```
   bash
   pm2 logs
   # or
   tail -f ~/.pm2/logs/*
   ```

2. **Run Diagnostics:**
   ```
   bash
   ./test_atlas_integration.sh
   ```

3. **Check Database:**
   ```bash
   sqlite3 prisma/dev.db ".tables"
   sqlite3 prisma/dev.db "SELECT * FROM AudioProcessor;"
   ```

4. **Verify Network:**
   ```bash
   ping 192.168.1.100
   nc -zv 192.168.1.100 5321
   telnet 192.168.1.100 5321
   ```

5. **Review Documentation:**
   - See `DIAGNOSTIC_AND_FIX_REPORT.md` for detailed analysis
   - Check `SYSTEM_DOCUMENTATION.md` for system overview

✅ Success Criteria

You'll know everything is working when:

1. ✅ Application loads without errors
2. ✅ Input gain sliders show values other than -40dB
3. ✅ Zone names match your Atlas configuration
4. ✅ Adjusting gains sends commands to Atlas
5. ✅ Zone controls respond immediately
6. ✅ No 500 errors in browser console
7. ✅ Logs show successful Atlas connections

📞 Support

For additional support:
- Review the comprehensive code analysis in `DIAGNOSTIC_AND_FIX_REPORT.md`
- Check existing documentation in the `docs/` directory
- Review recent commits related to Atlas fixes

Remember: The code is correct - focus on configuration and deployment!