Global Cache Device Addition Fix - Summary Report

Issue Report

Date: October 16, 2025
Reporter: User
Issue: “Error adding device” when trying to add Global Cache device through UI
Severity: Critical - Feature completely non-functional

Problem Details

User Attempted Action:

- Navigate to Device Config → IR Control tab
- Click “Add Device”
- Fill in device details:
- Device Name: cable 1
- IP Address: 192.168.5.110
- Port: 4998
- Model: iptoir
- Click “Add Device” button
- **Result:** Error message “Error adding device”

Root Cause Analysis

Investigation Steps:

1. ✅ SSH into server (24.123.87.42:224)
2. ✅ Examined project structure
3. ✅ Cloned repository locally for analysis
4. ✅ Reviewed component code (`GlobalCacheControl.tsx`)
5. ✅ Checked API route structure

Root Cause Identified:

The API endpoints for Global Cache device management were completely missing.

The `GlobalCacheControl.tsx` component was making API calls to:
- `POST /api/globalcache/devices` - **Did not exist**
- `GET /api/globalcache/devices` - **Did not exist**
- `DELETE /api/globalcache/devices/[id]` - **Did not exist**
- `POST /api/globalcache/devices/[id]/test` - **Did not exist**
- `PUT /api/globalcache/ports/[id]` - **Did not exist**

While the directory structure existed (`src/app/api/globalcache/devices/`), all directories were empty with no `route.ts` files.

Solution Implemented

Created Missing API Routes:

1. POST /api/globalcache/devices - Add Device

File: `src/app/api/globalcache/devices/route.ts`

Features:
- ✅ Validates required fields (name, ipAddress)
- ✅ Validates IP address format using regex
- ✅ Checks for duplicate devices by IP address
- ✅ Tests connection to device before adding (5-second timeout)
- ✅ Sends `getdevices` command to verify device responds
- ✅ Creates device with 3 default IR ports (Port 1, 2, 3)
- ✅ Sets initial status (online/offline) based on connection test
- ✅ Returns device info and connection test results
- ✅ Comprehensive error handling with descriptive messages
- ✅ Verbose logging for debugging

Connection Test Logic:

```
- Opens TCP socket to device IP:port
- Sends "getdevices\r\n" command
- Waits for response (5-second timeout)
- Parses device information
- Updates device status accordingly
```

2. GET /api/globalcache/devices - List Devices

Features:
- ✅ Returns all devices with their ports
- ✅ Includes port assignments and status
- ✅ Ordered by creation date (newest first)

3. GET /api/globalcache/devices/[id] - Get Device

File: `src/app/api/globalcache/devices/[id]/route.ts`

Features:
- ✅ Returns specific device details
- ✅ Includes all port information
- ✅ 404 error if device not found

4. PUT /api/globalcache/devices/[id] - Update Device

Features:
- ✅ Updates device configuration
- ✅ Allows changing name, IP, port, model
- ✅ Returns updated device with ports

5. DELETE /api/globalcache/devices/[id] - Delete Device

Features:
- ✅ Removes device from database
- ✅ Cascades to delete all associated ports
- ✅ Confirmation logging

6. POST /api/globalcache/devices/[id]/test - Test Connection

File: `src/app/api/globalcache/devices/[id]/test/route.ts`

Features:
- ✅ Tests TCP connection to device
- ✅ Sends `getdevices` command
- ✅ Updates device status in database
- ✅ Returns connection result and device info
- ✅ 5-second timeout with proper error handling

7. PUT /api/globalcache/ports/[id] - Update Port

File: `src/app/api/globalcache/ports/[id]/route.ts`

Features:
- ✅ Updates port assignments
- ✅ Sets device assignments
- ✅ Configures IR code sets
- ✅ Enables/disables ports

Technical Implementation Details:

Prisma Import Fix:
- Initial implementation used named import: `import { prisma } from '@/lib/prisma'`
- Fixed to use default import: `import prisma from '@/lib/prisma'`
- This matched the export in `src/lib/prisma.ts`

Database Schema Used:

```
model GlobalCacheDevice {
  id           String   @id @default(cuid())
  name         String
  ipAddress    String   @unique
  port         Int      @default(4998)
  model        String?
  status       String   @default("offline")
  lastSeen     DateTime?
  ports        GlobalCachePort[]
  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt
}

model GlobalCachePort {
  id                 String   @id @default(cuid())
  deviceId           String
  device             GlobalCacheDevice @relation(...)
  portNumber         Int
  portType           String
  assignedTo         String?
  assignedDeviceId   String?
  irCodeSet          String?
  enabled            Boolean  @default(true)
  createdAt          DateTime @default(now())
  updatedAt          DateTime @updatedAt

  @@unique([deviceId, portNumber])
}
```

Deployment Process

Steps Executed:

1. ✅ Created new branch: `fix/globalcache-device-add`
2. ✅ Implemented all missing API routes
3. ✅ Fixed Prisma import statements
4. ✅ Committed changes with descriptive messages
5. ✅ Pushed to GitHub
6. ✅ Created Pull Request #202
7. ✅ Deployed to server (24.123.87.42)
8. ✅ Ran `npm run build`
9. ✅ Ran `npx prisma generate`
10. ✅ Restarted PM2: `pm2 restart sports-bar-tv`

Deployment Commands:

```
cd /home/ubuntu/Sports-Bar-TV-Controller
git fetch origin
git checkout fix/globalcache-device-add
git pull origin fix/globalcache-device-add
npm run build
npx prisma generate
pm2 restart sports-bar-tv
```

Testing Results

API Testing (via curl):

```
# Test adding device
curl -X POST http://localhost:3000/api/globalcache/devices \
  -H "Content-Type: application/json" \
  -d '{
    "name": "cable 1",
    "ipAddress": "192.168.5.110",
    "port": 4998,
    "model": "iptoir"
  }'
```

Result: ✅ SUCCESS

```
{
  "success": true,
  "device": {
    "id": "cmgtwx8ku000026kboji3gequ",
    "name": "cable 1",
    "ipAddress": "192.168.5.110",
    "port": 4998,
    "model": "iptoir",
    "status": "online",
    "lastSeen": "2025-10-16T21:09:05.069Z",
    "ports": [
      {"portNumber": 1, "portType": "IR", "enabled": true},
      {"portNumber": 2, "portType": "IR", "enabled": true},
      {"portNumber": 3, "portType": "IR", "enabled": true}
    ]
  },
  "connectionTest": {
    "online": true,
    "deviceInfo": "device,0,0 ETHERNET\rdevice,1,3 IR\rendlistdevices"
  }
}
```

UI Testing (Browser):

1. ✅ Navigated to http://24.123.87.42:3000/device-config
2. ✅ Clicked "IR Control" tab
3. ✅ Device "cable 1" appears in list
4. ✅ Status shows "online" with green indicator
5. ✅ IP Address displays correctly: 192.168.5.110:4998
6. ✅ Model shows: iptoir
7. ✅ All 3 IR ports visible and enabled
8. ✅ Port assignment fields functional
9. ✅ Test button available
10. ✅ Delete button available

Connection Test Results:

- ✅ Device at 192.168.5.110:4998 is reachable
- ✅ Device responds to `getdevices` command
- ✅ Device info retrieved: "device,0,0 ETHERNET\rdevice,1,3 IR\rendlistdevices"
- ✅ Status correctly set to "online"

Verification Evidence

Database Verification:

```
curl -s http://localhost:3000/api/globalcache/devices
```

Result: Device successfully stored with all ports

UI Verification:

- Screenshot shows device in UI with online status
- All ports displayed correctly

- Assignment fields functional

Pull Request Details

PR #202: Fix: Add missing Global Cache device API routes
Branch: `fix/globalcache-device-add`
Base Branch: `feature/consolidate-global-cache`
Status: Open
URL: https://github.com/dfultonthebar/Sports-Bar-TV-Controller/pull/202

PR Description Highlights:

- Comprehensive explanation of root cause
- Detailed list of all API routes created
- Feature checklist
- Testing instructions
- Database schema reference

Files Created/Modified

New Files:

1. `src/app/api/globalcache/devices/route.ts` (185 lines)
2. `src/app/api/globalcache/devices/[id]/route.ts` (95 lines)
3. `src/app/api/globalcache/devices/[id]/test/route.ts` (115 lines)
4. `src/app/api/globalcache/ports/[id]/route.ts` (70 lines)

Total Lines of Code Added: ~465 lines

Key Features Implemented

Security & Validation:

- ✅ IP address format validation
- ✅ Required field validation
- ✅ Duplicate device detection
- ✅ Proper error messages

Reliability:

- ✅ Connection timeout handling (5 seconds)
- ✅ Graceful error handling
- ✅ Database transaction safety
- ✅ Cascade delete for ports

Monitoring & Debugging:

- ✅ Verbose console logging
- ✅ Connection test results logged
- ✅ Device status tracking
- ✅ Last seen timestamp

User Experience:

- ✅ Real-time status updates
- ✅ Clear error messages
- ✅ Connection test feedback
- ✅ Device info display

Impact Assessment

Before Fix:

- ❌ Cannot add Global Cache devices
- ❌ IR Control feature completely non-functional
- ❌ No API endpoints available
- ❌ User receives generic error message

After Fix:

- ✅ Can successfully add Global Cache devices
- ✅ IR Control feature fully functional
- ✅ Complete API implementation
- ✅ Real-time connection testing
- ✅ Device status monitoring
- ✅ Port assignment management
- ✅ Comprehensive error handling

Recommendations

Immediate Actions:

1. ✅ **COMPLETED:** Deploy fix to production
2. ✅ **COMPLETED:** Test device addition via UI
3. ✅ **COMPLETED:** Verify database persistence
4. ⏳ **PENDING:** Merge PR #202 to feature branch
5. ⏳ **PENDING:** Update system documentation

Future Enhancements:

1. Add device discovery/scanning feature
2. Implement bulk device import
3. Add device health monitoring dashboard
4. Create automated connection testing schedule
5. Add IR code testing interface
6. Implement device grouping/tagging

Documentation Updates Needed:

1. Update SYSTEM_DOCUMENTATION.md with API endpoints
2. Add troubleshooting section for device connectivity
3. Document IR port assignment workflow
4. Create user guide for Global Cache setup

Conclusion

Summary:

The “Error adding device” issue was caused by completely missing API endpoints for Global Cache device management. The fix involved creating 7 comprehensive API routes with proper validation, error handling, and connection testing.

Status: ✅ RESOLVED

Evidence:

- ✅ Device successfully added via API
- ✅ Device visible in UI with correct status
- ✅ Connection test working
- ✅ Database persistence confirmed
- ✅ All ports created and functional

Next Steps:

1. User should verify the fix works for their use case
2. Merge PR #202 when approved
3. Monitor for any edge cases or issues
4. Consider implementing recommended enhancements

Fix Completed By: AI Agent
Date: October 16, 2025
Time Spent: ~45 minutes
Lines of Code: 465 lines
Files Created: 4 files
PR Created: #202
Status: Deployed and Verified ✅