Remote Control Popup Feature Documentation

Overview

The Remote Control Popup feature provides context-aware remote controls for different device types (Amazon Fire TV, DirecTV, and Cable Box) directly within the Bartender Channel Guide interface.

Features

1. Context-Aware Remote Display

- Automatically detects the device type based on the selected input
- Shows the appropriate remote control UI for the device
- Three device types supported:
- **Amazon Fire TV** (Streaming)
- **DirecTV** (Satellite)
- **Cable Box** (IR/Global Cache)

2. Remote Control Components

Fire TV Remote (`src/components/remotes/FireTVRemote.tsx`)

Features:
- Navigation controls (Up, Down, Left, Right, Select)
- Home, Back, and Menu buttons
- Playback controls (Play/Pause, Rewind, Fast Forward)
- Volume controls (Vol+, Vol-, Mute)
- Visual feedback for button presses
- Loading indicators during command execution

API Integration:
- Endpoint: `/api/firetv-devices/send-command`
- Uses ADB over IP for command execution
- Persistent connection management

DirecTV Remote (`src/components/remotes/DirecTVRemote.tsx`)

Features:
- Full navigation pad with OK button
- Number pad for direct channel entry (0-9)
- Channel controls (CH+, CH-, Last Channel)
- Guide, Menu, List, and Info buttons
- DVR controls (Play, Pause, Record, Stop, Rewind, FF)
- Skip controls (Skip Back -30s, Skip Forward +30s)
- Volume controls with Mute
- Channel input display with auto-entry
- Back and Exit buttons

API Integration:
- Endpoint: `/api/directv-devices/send-command`
- Uses DirecTV SHEF (Serial Home Electronics Framework) API
- HTTP-based commands over IP

Cable Box Remote (`src/components/remotes/CableBoxRemote.tsx`)

Features:
- Power button
- Navigation controls with OK button
- Number pad for channel entry
- Channel controls (CH+, CH-, Last)
- Guide, Menu, and Info buttons
- DVR/Playback controls
- Skip controls (Skip Back -10s, Skip Forward +30s)
- Record button
- Volume controls with Mute
- Auto-submit after 3 digits for channel entry

API Integration:
- Endpoint: `/api/ir-devices/send-command`
- Uses Global Cache iTach IR transmitters
- Sends infrared commands to control cable boxes

3. RemoteControlPopup Component (`src/components/remotes/RemoteControlPopup.tsx`)

Purpose:
Main container component that manages the modal overlay and routes to the appropriate remote control UI based on device type.

Features:
- Modal overlay with backdrop blur
- ESC key to close
- Click outside to dismiss
- Prevents body scroll when open
- Close button (X) in top-right corner
- Responsive design for desktop and tablet

4. Integration with Channel Guide

Trigger:
- “Remote Control” button appears in the header when a device is selected
- Button is styled with blue accent color and gamepad icon
- Only visible when `selectedDevice` is not null

Integration Points:
- Added to `EnhancedChannelGuideBartenderRemote.tsx`
- Uses existing device state management
- Leverages existing device type detection (`getDeviceTypeForInput()`)
- Integrates with existing logging system

Usage

For End Users

1. Select an input from the left panel in the Bartender Channel Guide
2. Click the “Remote Control” button in the header
3. Use the on-screen remote to control the selected device
4. Click the X button, press ESC, or click outside to close

For Developers

Adding New Remote Commands

To add new commands to any remote:

1. Update the remote component (e.g., `FireTVRemote.tsx`)
2. Add the button to the UI
3. Call `sendCommand()` with the appropriate command string
4. The API endpoint handles command mapping and execution

Modifying Remote Layout

Each remote component is self-contained:
- Uses Tailwind CSS for styling
- Uses Lucide React icons
- Grid-based layouts for button arrangement
- Consistent color scheme across all remotes

API Endpoints

Fire TV Commands

```
POST /api/firetv-devices/send-command
Body: {
  deviceId: string
  command: string (UP, DOWN, LEFT, RIGHT, OK, HOME, BACK, MENU, etc.)
  ipAddress: string
  port: number
}
```

DirecTV Commands

```
POST /api/directv-devices/send-command
Body: {
  deviceId: string
  command: string (UP, DOWN, LEFT, RIGHT, OK, GUIDE, MENU, etc.)
  ipAddress: string
  port: number
}
```

IR/Cable Box Commands

```
POST /api/ir-devices/send-command
Body: {
  deviceId: string
  command: string (POWER, UP, DOWN, CH_UP, VOL_UP, etc.)
  iTachAddress: string
}
```

Testing

Manual Testing Checklist

- [ ] Fire TV Remote
- [ ] Navigation controls work
- [ ] Home/Back/Menu buttons respond
- [ ] Playback controls function
- [ ] Volume controls operate correctly
- [ ] DirecTV Remote
- [ ] Navigation and OK button work
- [ ] Number pad for channel entry
- [ ] Channel controls (up/down/last)
- [ ] DVR controls (play/pause/record)
- [ ] Guide and Menu buttons
- [ ] Cable Box Remote
- [ ] Power button works
- [ ] Navigation controls respond
- [ ] Number pad channel entry
- [ ] IR commands sent successfully
- [ ] Volume controls operate

UI/UX Testing

- [ ] Modal opens smoothly
- [ ] ESC key closes modal
- [ ] Click outside dismisses modal
- [ ] Visual feedback on button press
- [ ] Loading indicators display
- [ ] Error messages show appropriately
- [ ] Success messages confirm actions
- [ ] No body scroll when modal is open
- [ ] Responsive on different screen sizes

Known Limitations

1. **IR Commands**: The cable box remote uses predefined IR codes. Custom codes may need to be added to the database for specific cable box models.

2. **Network Connectivity**: All remotes require network connectivity to the devices. If a device is offline, commands will fail with appropriate error messages.

3. **DirecTV SHEF**: DirecTV receivers must have External Device Access enabled in their settings for network control to work.

4. **Fire TV ADB**: Fire TV devices must have ADB debugging enabled and be authorized for the control system.

Future Enhancements

1. **Macro Support**: Add ability to create and execute command macros (e.g., “Watch ESPN” = power on + change input + tune to channel).

2. **Customizable Layouts**: Allow users to customize remote button layouts per device.

3. **Learning Mode**: For IR devices, add ability to learn new commands from physical remotes.

4. **Voice Control**: Integrate voice commands for hands-free operation.

5. **Multi-Device Control**: Allow simultaneous control of multiple devices (e.g., TV + Cable Box).

6. **Favorite Commands**: Quick access to frequently used commands.

Deployment

Build and Deploy

```
# Build the application
npm run build

# Test build locally
npm run start

# Deploy to production (via SSH)
ssh -p 224 ubuntu@24.123.87.42
cd /path/to/app
git pull
npm install
npm run build
pm2 restart sports-bar-app
```

Production Checklist

- [ ] All TypeScript errors resolved
- [ ] Build completes successfully
- [ ] No console errors in browser
- [ ] Test with real devices on production server
- [ ] Verify all three device types work
- [ ] Check error handling and logging

- [ ] Confirm responsive design on tablets

Troubleshooting

Remote Not Appearing

- Check that a device is selected (selectedDevice is not null)
- Verify the device has a valid device type (cable/satellite/streaming)

Commands Not Working

- **Fire TV**: Check ADB connection, verify device IP and port
- **DirecTV**: Ensure SHEF is enabled, check IP and port 8080
- **Cable Box**: Verify iTach address, check IR codes in database

UI Issues

- Clear browser cache
- Check console for JavaScript errors
- Verify all components are imported correctly

Support

For issues or questions:
1. Check console logs for error messages
2. Verify device connectivity and settings
3. Review API endpoint responses
4. Check database for device configurations

Version History

v1.0.0 (October 28, 2025)

- Initial release
- Fire TV, DirecTV, and Cable Box remote controls
- Modal popup interface
- Integration with Bartender Channel Guide
- Context-aware device detection