

MANAR LABS

LiveDeck

Control API Reference

Every local HTTP route LiveDeck's iOS, Mac, Apple TV, Android and Android TV apps expose for Bitfocus Companion, Stream Deck, and your own scripts — traced to source, with platform differences called out rather than hidden.

Version 1.0

Published 2 October 2026

Routes documented 71 (iOS/Mac/Apple TV + Android/Android TV)

Online version manarlabs.com/apps/livedeck/api/

Contents

Overview — ports, auth, discovery, CORS, rate limits

Pairing	2 routes
State & monitoring	14 routes
Run a Show	4 routes
Satellite	4 routes
Switching	6 routes
Skins	5 routes
PTZ	5 routes
Audio	10 routes
Outputs & Go Live	11 routes
NDI network	2 routes
Raw command	1 route
TV routes	7 routes
Errors & status codes	
Changelog	

Overview

Ports

iOS / iPadOS / Mac / Apple TV	8088
Android / Android TV	8090

Both are user-configurable in the app (Settings → Output → Web), within range 1024–65535.

Authentication

Every `/api/*` route except the two pairing routes requires an API token, sent either as an `Authorization: Bearer <token>` header or a `?token=` query parameter. An empty/unconfigured token closes the API entirely (every route answers 401/403). This token is separate from the human web remote's PIN.

Find the token — iOS/Mac/TV Output tab → Control API card → Show/Copy/Regenerate

Find the token — Android Output tab → Control API card → Show/Copy/Regenerate (same card appears under TV Mode on Android TV)

Pairing without typing a token

`POST /api/pair` and `GET /api/pair/<id>` are the only two unauthenticated `/api/*` routes — and they only answer a peer already on the same local network. See the **Pairing** section for the full exchange. This is also the only way to pair an Apple TV, which has no keyboard.

Discovery — Bonjour / mDNS

Both platforms advertise the API over mDNS as `_livedeck-api._tcp`, with the port in a TXT record — this is how Bitfocus Companion and this reference's own "Try it on your device" box find a device automatically on the same network.

CORS

Both the iOS/Mac/tvOS server and the Android/Android TV server send `Access-Control-Allow-Origin: *` on every response, including every `/api/*` route and the SSE stream — confirmed directly in source (`WebControlServer.swift`, `WebControl.kt`). A browser page opened from local disk (`file://`, `Origin null`) can call the device directly with `fetch()`; no proxy and no app change were required for the downloadable standalone tester.

Rate limits

Only the pairing routes are rate-limited: `POST /api/pair` allows 5 requests/minute per peer (429 past that) and one pending pairing request at a time (409). The token-gated `/api/*` routes are not themselves rate-limited beyond normal engine processing.

Local network only

This is a local HTTP server with no internet-facing component — nothing you send to these routes passes through Manar Labs' servers. Reach a device only over the same Wi-Fi/LAN (or a VPN that bridges it), by IP or `localhost` hostname.

Pairing (2)

The only two /api routes that don't need a token — but they only answer a peer on the same local network. Use this to get a token into a browser or a Companion install without ever showing the token on screen.

POST

/api/pair

No token (local network only)

PARAM	WHERE	REQUIRED	DESCRIPTION
-------	-------	----------	-------------

name	body (JSON)	Yes	Label shown in the app's pairing prompt, ≤40 characters.
------	-------------	-----	--

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/pair \  
-H 'Content-Type: application/json' \  
-d '{"name": "My browser"}'
```

EXAMPLE RESPONSE

```
{ "requestID": "a1b2c3", "code": "4821", "expiresIn": 60 }
```

Rate-limited to 5 requests/min per peer (429 past that), one pending pair at a time (409), and 503 if the app has no token configured yet (API fully closed). The app shows the 4-digit code for the operator to approve or deny on-screen.

GET

/api/pair/<requestID>

No token (local network only)

PARAM	WHERE	REQUIRED	DESCRIPTION
-------	-------	----------	-------------

requestID	path	Yes	The id returned by POST /api/pair.
-----------	------	-----	------------------------------------

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/pair/a1b2c3
```

EXAMPLE RESPONSE

```
{ "state": "waiting", "expiresIn": 42 }  
// then, once approved:  
{ "state": "approved", "token": "<48-char hex token>" }  
// or: { "state": "denied" } / { "state": "expired" }
```

Poll this every couple of seconds. The token is handed over exactly once — the server destroys the pairing record right after a successful "approved" read, so a second poll after that returns "expired".

State & monitoring (14)

Read-only. Everything here works with GET and never changes anything in the app.

GET /api/state

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/state?token={token}
```

EXAMPLE RESPONSE

Full state JSON – the same object the browser remote polls as /state.json.

GET /api/program

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/program?token={token}
```

EXAMPLE RESPONSE

```
{ "program": "...", "liveLayout": "...", "cuedLayout": "...", "takeStyle": "cut",  
  "portrait": false, "programPtz": {...} }
```

Android's response additionally includes "preview" and "monMuted" keys that iOS does not return here.

GET /api/layouts

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/layouts?token={token}
```

EXAMPLE RESPONSE

Array of layout objects (id, name, sources, skins...).

GET /api/sources

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/sources?token={token}
```

EXAMPLE RESPONSE

Array of source objects currently known to the mixer (NDI, cameras, satellites, test patterns).

POST**/api/sources/<id>/remove**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id, from /api/sources.

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/sources/cam1/remove?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/skins**

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/skins?token={token}
```

EXAMPLE RESPONSE

```
iOS: { skins, skinKinds, skinEffects, skinTriggers, skinGroups, layouts }
```

Android's response is narrower: only { skins, skinKinds }. The skinEffects/skinTriggers/skinGroups/per-layout skin list are iOS-only on this route.

GET**/api/transitions**

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/transitions?token={token}
```

EXAMPLE RESPONSE

```
Array of available transition styles (cut, dissolve, ...).
```

GET**/api/audio**

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio?token={token}
```

EXAMPLE RESPONSE

```
Per-source audio state: gain, mute, solo, delay, follow mode, master level.
```

GET /api/health

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/health?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Cheap liveness check — good for a Companion "device reachable" indicator.

GET /api/stats

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/stats?token={token}
```

EXAMPLE RESPONSE

```
Engine stats: frame rate, dropped frames, CPU/thread load, output bitrates.
```

GET /api/events

SSE stream

Bearer token

EXAMPLE REQUEST

```
curl -N http://{ip}:{port}/api/events?token={token}
```

EXAMPLE RESPONSE

```
text/event-stream. `event: state` with a compact state payload on every change, plus a `:ping` comment every ~15s to keep the connection alive.
```

iOS's compact payload includes isLive/liveSince/liveSelection; Android's compact-key set omits those three fields. Not fetch-able from the standalone tester (EventSource keeps the connection open) — use curl -N or a Companion feedback instead.

GET`/api/thumb/<id>`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/thumb/cam1?token={token} -o thumb.jpg
```

EXAMPLE RESPONSE

```
image/jpeg
```

Hitting this also counts as "someone is watching" and keeps that source's thumbnail refreshing.

GET`/api/program.jpg`

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/program.jpg?token={token} -o program.jpg
```

EXAMPLE RESPONSE

```
image/jpeg - a snapshot of the current Program output.
```

GET`/api/outputs`

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/outputs?token={token}
```

EXAMPLE RESPONSE

```
iOS: { out: { ndi, rtmp, record, hls, rtmpSrv, ... }, isLive, liveSince, liveSelection,
goLiveRemembered, lastGoLiveFailures }
Android: { out: { ndi, rec, rtmp, rtmpStatus, fb, fbStatus, fbConfigured, hls, hlsURL,
viewers } }
```

The two platforms' output objects genuinely differ: Android exposes native Facebook Live fields (fb/fbStatus/fbConfigured) that iOS does not have via this API; iOS exposes GO LIVE bookkeeping (isLive/liveSince/liveSelection/goLiveRemembered) that Android's table doesn't list here.

Run a Show (4)

Starts, steps through and ends a saved rundown (intro/outro, cued segments).

POST

/api/show/start

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
record	query	No	Default true.
stream	query	No	Default true.

EXAMPLE REQUEST

```
curl -s -X POST 'http://{ip}:{port}/api/show/start?token={token}&record=true&stream=true'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST

/api/show/next

Bearer token

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/show/next?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST

/api/show/step/<index>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
index	path	Yes	0-based step index.

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/show/step/2?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST

/api/show/end

Bearer token

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/show/end?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Satellite (4)

Controls a connected Satellite (a second phone sending camera/screen/test pattern) from the Studio side.

POST /api/satellite/talk/<on>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
on	path	Yes	"on" or "off".

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/satellite/talk/on?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST /api/satellite/talk/target/<target>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
target	path	Yes	"auto", "all", or a source id.

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/satellite/talk/target/auto?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST /api/satellite/camera/<action>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
action	path	Yes	"flip", "torch", or "zoom".
value	query	No	Used by "zoom".
to	query	No	Target satellite, if more than one.

EXAMPLE REQUEST

```
curl -s -X POST 'http://{ip}:{port}/api/satellite/camera/flip?token={token}'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST**/api/satellite/message****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
text	query/body	Yes	Message text.
to	query	No	Target satellite, if more than one.

EXAMPLE REQUEST

```
curl -s -X POST 'http://{ip}:{port}/api/satellite/message?token={token}&text=Go+to+camera+2'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Switching (6)

Cue and take layouts — the core vision-mixing actions.

GET /api/layout/<id>/cue

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Layout id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/layout/lower-third/cue?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Cues green — does not go to air.

GET /api/layout/<id>/take

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Layout id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/layout/lower-third/take?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Takes to air with the studio's current default transition.

GET /api/layout/<id>/take/<style>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Layout id.
style	path	Yes	Transition style, e.g. cut, dissolve.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/layout/lower-third/take/cut?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/take****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
style	query	No	Optional transition style override.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/take?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Takes the currently cued layout.

GET**/api/cut****Bearer token****EXAMPLE REQUEST**

```
curl -s http://{ip}:{port}/api/cut?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Shortcut for take with a hard cut.

GET**/api/transition/<style>****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
style	path	Yes	New studio default transition style.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/transition/dissolve?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Skins (5)

Lower thirds, tickers, clocks and other overlays played over Program.

GET

/api/skin/<id>/play

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Skin id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/skin/ticker1/play?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET

/api/skin/<id>/stop

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Skin id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/skin/ticker1/stop?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST

/api/skin/<id>/color

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Skin id.
value	query	Yes	#RRGGBB, or "original" to reset.

EXAMPLE REQUEST

```
curl -s -X POST 'http://{ip}:{port}/api/skin/ticker1/color?token={token}&value=%23FF3366'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/catalog****Bearer token****EXAMPLE REQUEST**

```
curl -s http://{ip}:{port}/api/catalog?token={token}
```

EXAMPLE RESPONSE

```
Merged bundled + remote skin catalogue index.
```

```
Served directly, not routed through the generic command table.
```

GET**/api/catalog/<id>/add****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Catalogue item id.
layoutId	query	No	Layout to add the skin to.

EXAMPLE REQUEST

```
curl -s 'http://{ip}:{port}/api/catalog/lower-third-01/add?token={token}&layoutId=main'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

PTZ (5)

Pan/tilt/zoom cameras — presets, direct moves and storing new presets.

GET /api/ptz/cameras

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/ptz/cameras?token={token}
```

EXAMPLE RESPONSE

```
Array of PTZ-capable camera sources.
```

GET /api/ptz/<target>/preset/<n>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
target	path	Yes	Camera source id.
n	path	Yes	Preset number.
speed	query	No	Android only — default "1".

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/ptz/cam1/preset/3?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Android's handler also sends an optional speed parameter (default "1") that iOS's route does not declare.

GET /api/ptz/<target>/store/<n>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
target	path	Yes	Camera source id.
n	path	Yes	Preset number to store the current position as.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/ptz/cam1/store/3?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/ptz/<target>/pantilt****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
target	path	Yes	Camera source id.
pan	query	Yes	Pan speed/position.
tilt	query	Yes	Tilt speed/position.

EXAMPLE REQUEST

```
curl -s 'http://{ip}:{port}/api/ptz/cam1/pantilt?token={token}&pan=0.2&tilt=-0.1'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/ptz/<target>/zoom****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
target	path	Yes	Camera source id.
speed	query	Yes	Zoom speed.

EXAMPLE REQUEST

```
curl -s 'http://{ip}:{port}/api/ptz/cam1/zoom?token={token}&speed=0.5'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Audio (10)

Per-source gain, mute, solo, delay and master level.

GET /api/audio/<id>/delay/<ms>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.
ms	path	Yes	0–500.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/delay/40?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET /api/source/<id>/videodelay/<ms>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.
ms	path	Yes	0–500.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/source/cam1/videodelay/40?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET /api/audio/master/<value>

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
value	path	Yes	0...1 on Android; iOS allows slightly over 1 for makeup gain.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/master/0.8?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/audio/mode/<mode>`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
mode	path	Yes	"follow" or "manual".

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/mode/follow?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/audio/<id>/gain/<value>`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.
value	path	Yes	Gain level.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/gain/0.9?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/audio/<id>/mute`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/mute?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Toggles.

GET**/api/audio/<id>/solo****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/solo?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Toggles.

GET**/api/audio/<id>/delete****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/delete?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/audio/<id>/follow/<layoutId>****Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
id	path	Yes	Source id.
layoutId	path	Yes	Layout to follow.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/audio/cam1/follow/main?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/monitor/mute/<state>`**Bearer token**

PARAM	WHERE	REQUIRED	DESCRIPTION
state	path	Yes	"on" or "off".

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/monitor/mute/on?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Outputs & Go Live (11)

Starts/stops NDI, RTMP, recording and streaming outputs, and the "GO LIVE" shortcut that starts everything ticked at once.

GET /api/remote/pin

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/remote/pin?token={token}
```

EXAMPLE RESPONSE

```
{ "pin": "4821" }
```

This is the human web-remote's PIN, unrelated to the API bearer token.

POST /api/remote/pin

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
pin	query/body	Yes	4–8 digits, or "new" for a random one.

EXAMPLE REQUEST

```
curl -s -X POST 'http://{ip}:{port}/api/remote/pin?token={token}&pin=new'
```

EXAMPLE RESPONSE

```
{ "pin": "7310" }
```

POST /api/output/nas

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
host	body	Yes	
share	body	Yes	
folder	body	No	
user	body	No	
password	body	No	

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/output/nas?token={token} \
-d host=192.168.1.50 -d share=Recordings
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Configures record-to-NAS.

POST`/api/output/nas/clear`

Bearer token

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/output/nas/clear?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/output/ndi/name/<name>`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
name	path	Yes	New NDI output name.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/output/ndi/name/Studio-A?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET`/api/output/quality/<quality>`

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
quality	path	Yes	iOS: 480p 720p 1080p. Android: 720p 1080p.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/output/quality/1080p?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

iOS accepts 480p; Android does not offer it.

GET**/api/output/portrait/<on>**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
on	path	Yes	"on" or "off".

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/output/portrait/on?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/output/<name>/<state>**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
name	path	Yes	iOS: ndi rtmp record hls rtmpserver. Android: ndi record rtmp facebook hls.
state	path	Yes	"on" or "off".
url	query	No	Android only, for name=rtmp — start with a specific server URL.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/output/ndi/on?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

The accepted name vocabulary differs by platform: rtmpserver exists only on iOS, facebook only on Android. Android's own /api/golive/select route (below) uses the iOS vocabulary instead of this one's — see the note there.

POST**/api/golive**

Bearer token

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/golive?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Starts every output currently ticked in Output settings.

POST**/api/golive/stop**

Bearer token

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/golive/stop?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Confirm before wiring this to a hotkey — it stops every live output at once.

GET**/api/golive/select/<name>/<state>**

Bearer token

PARAM WHERE REQUIRED DESCRIPTION

name	path	Yes	ndi rtmp record hls rtmpserver — the same set on both platforms.
state	path	Yes	"on" or "off" — ticks/unticks it for the next GO LIVE.

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/golive/select/rtmp/on?token={token}
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

On Android this route's name vocabulary (rtmpserver, no facebook) differs from /api/output/<name>/<state>'s own vocabulary (facebook, no rtmpserver) on the same device — flagged to engineering, not something this page can paper over.

NDI network (2)

Configures how this device finds NDI sources — a discovery server address and/or a list of direct IPs.
Answers on every device regardless of mode.

GET /api/ndi/network

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/ndi/network?token={token}
```

EXAMPLE RESPONSE

```
{ "discoveryServer": "", "directIPs": [] }
```

POST /api/ndi/network

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
discoveryServer	body	No	
directIPs	body	No	Comma-separated. Either/both fields; empty clears.

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/ndi/network?token={token} \
-d directIPs=192.168.1.21,192.168.1.22
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Raw command (1)

The generic forwarder behind the browser remote's own control panel. Covers everything that doesn't have a dedicated shortcut route above.

POST

/api/command

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
action	body or query, plus any action-specific fields	Yes	e.g. addLayout, renameLayout, duplicateLayout, removeLayout, addSkin, updateSkin, removeSkin, playSkin, stopSkin, camZoom, testPattern, addInput, addCamera, ...

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/command?token={token} \
-H 'Content-Type: application/json' \
-d '{"action":"testPattern","on":true}'
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET is also accepted with the same params on the query string. This is the same JSON shape the browser remote's own POST /cmd uses.

TV routes (7)

Only answered when the device is running in TV box mode (Apple TV / Android TV). On a phone, tablet or Mac these fall through to the generic 404.

GET /api/tv

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/tv?token={token}
```

EXAMPLE RESPONSE

```
Current TV mode + status.
```

POST /api/tv/mode

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
-------	-------	----------	-------------

mode	query/body	Yes	watch studio choose (iOS also documents "choose" explicitly to return to the chooser screen).
------	------------	-----	---

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/tv/mode?token={token}&mode=watch
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST /api/tv/fullscreen

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
-------	-------	----------	-------------

on	query/body	Yes	
----	------------	-----	--

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/tv/fullscreen?token={token}&on=true
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST**/api/tv/autostart**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
on	query/body	Yes	

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/tv/autostart?token={token}&on=true
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

POST**/api/tv/quality**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
auto	query/body	Yes	

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/tv/quality?token={token}&auto=true
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

GET**/api/tv/sources**

Bearer token

EXAMPLE REQUEST

```
curl -s http://{ip}:{port}/api/tv/sources?token={token}
```

EXAMPLE RESPONSE

```
Sources the TV box can watch.
```

POST**/api/tv/watch**

Bearer token

PARAM	WHERE	REQUIRED	DESCRIPTION
source	query/body	Yes	Source id to watch.

EXAMPLE REQUEST

```
curl -s -X POST http://{ip}:{port}/api/tv/watch?token={token}&source=cam1
```

EXAMPLE RESPONSE

```
{ "ok": true }
```

Errors & status codes

Consolidated from the behaviour documented against individual routes above.

STATUS	MEANING
401 / 403	Missing, wrong, or unconfigured API token (an app with no token set closes the API entirely). 403 also covers pairing requests from off-LAN.
404	Unknown route — or, on a device running in TV-only box mode, any non-TV <code>/api/*</code> route (the response points at <code>/api/tv/mode</code>).
409	POST <code>/api/pair</code> while a pairing request is already pending.
429	POST <code>/api/pair</code> past 5 requests/minute from the same peer.
503	POST <code>/api/pair</code> when the app has no API token configured yet.

Changelog

VERSION DATE		NOTES
1.0	2 October 2026	Initial public reference — 71 routes traced to source across iOS/Mac/Apple TV and Android/Android TV, generated from the same data as manarlabs.com/apps/livedeck/api/ .