DjMirrorPulse Local AI API (1.0.0)

Download OpenAPI specification: Download

Uniform contract for every `/api/v1/*` route exposed by DjMirrorPulse's HTTP configuration server (`HttpConfigServer`). Every response is wrapped in the same envelope (`{ok, data, error, meta}`) where `meta.revision` is a monotonic live-config counter. Exceptions: `GET /config/subscribe` (long-poll event shape) and binary download routes (raw bytes, no envelope).

This file is the single source of truth for the `/api/v1` surface: every route the server registers has an operation here, each with a unique `operationId`.
`docs_tech/docs_http_ws/README_HTTP_API_REFERENCE.md` keeps the design notes. The legacy, unversioned surface (ad-hoc per-endpoint shapes) is intentionally out of scope here; with an API key set it needs the same credentials as /api/v1.

Vendor extensions used in this file:

- `x-edition` (operation): the minimum license edition (`professional`, `enterprise`) needed to switch the feature on or start it. Reading its state always works; a refused call answers 403 EDITION_REQUIRED. The editions are listed by `GET /license`.
- `x-internal` (operation): not part of the supported contract (test-only routes); leave them out of published documentation.
- `x-websocket-events` (root): the events pushed over the WebSocket, which OpenAPI cannot describe.

Authentication is the user's choice: with no API key ("apiKey" in config.json, or DJMIRROR_API_TOKEN, which wins) every route answers without credentials. With one, every route needs the API key or the deviceToken of a phone paired by QR, except the pairing routes that declare their own security below.

WebSocket events

Pushed over ws://localhost:<httpPort+1> (with an API key, add ?token=). Every event except config.changed arrives as {type: event, eventType: custom.event, resource: event, action: triggered, name, data, requestId, timestamp}; config.changed uses eventType config.changed with resource config and action updated. These events are not delivered to the GET /config/subscribe long-poll.

Event	When	`data`
`config.changed`	The live config changed. Same body as the long-poll event; also pushed over the WebSocket as eventType config.changed (not custom.event).	the changed fragment
`ping`	Heartbeat every 30 s while at least one client is connected.	none
`license.changed`	The edition in force changed (a pairing or activation changed it).	LicenseInfo-like object
`recording.started`	A recording started.	as POST /recording/start
`recording.stopped`	A recording stopped.	as POST /recording/stop
`recording.snapshot`	A snapshot was captured.	as POST /recording/snapshot
`spout.started`	The Spout output started.	SpoutStatus
`spout.stopped`	The Spout output stopped.	SpoutStatus
`spout.updated`	Its settings changed (at most once per 150 ms).	SpoutConfig
`spout.error`	The output failed (for instance a source that gives no texture).	{code, message}
`spout.layout.started`	The Spout layout service started.	SpoutLayoutStatus
`spout.layout.stopped`	The Spout layout service stopped.	SpoutLayoutStatus
`spout.layout.updated`	Its settings or the reloaded file changed (at most once per 150 ms).	SpoutLayoutConfig
`spout.layout.error`	A service-level failure, such as a bad layout file. One output's own failure only shows in the status.	{code, message}
`ndi.started`	The NDI bridge started.	NdiStatus

Event	When	`data`
`ndi.stopped`	The NDI bridge stopped.	NdiStatus
`ndi.updated`	Its settings changed (at most once per 150 ms).	NdiSettings
`ndi.error`	The bridge exited on its own while expected to keep running.	{code: EXITED, message}
`dmx.fixtures.updated`	The fixture list changed on disk or through the API.	none
`dmx.scenes.updated`	The scenes changed.	none
`dmx.fixtureProfiles.updated`	The fixture profiles changed.	none
`dmx.movementPatterns.updated`	The movement patterns changed.	none
`dmx.laserPatterns.updated`	The laser patterns changed.	none
`dmx.config.updated`	The DMX configuration changed.	none
`dmx.runtime.changed`	Live DMX output changed (scene applied, channels set).	optional, event-specific
`dmx.blackout.changed`	Blackout switched.	{active}
`dmx.silence.changed`	The silence state or its fade settings changed.	{active, fadeMs, curve}
`dmx.macro.triggered`	A macro was triggered.	event-specific
`dmx.playall.change`	Play All switched (POST /dmx/runtime/playall).	{running}

Event	When	`data`
`d`		
`dmx.scenemode.changed`	The forced scene mode changed (POST /dmx/runtime/scene-mode).	{mode}
`dmx.autoconnect.changed`	The DMX auto-connect setting changed.	event-specific

Health

Liveness and capability discovery.

Liveness check

GET `/health`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

- `ok` required
  boolean
  Value: `true`
- `data` required
  object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required
  object or null
  Always null on success.

meta > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 | 401

Content type: application/json

Copy | Expand all | Collapse all


```
{
    "ok": true,
  - "data": {
        "status": "ok",
        "realtimeWsPort": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Server capability discovery

GET /capabilities

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "apiVersion": "v1",
        "auth": "bearer_or_x_api_key",
        "supportsLongPoll": true,
        "supportsWebSocket": true,
        "routesPrefix": "/api/v1",
      + "spout": { … }
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the active license edition and its features

GET /license

The edition in force on this computer and, for every feature that is not in all editions, whether it is included. A paired phone sees the desktop's license.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data > required	object Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "edition": "professional",
        "isActive": true,
        "isTrial": false,
        "trialDaysRemaining": 0,
      + "features": { … }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Config

Generic and flattened "global settings" configuration access.

Get the full config tree

GET /config

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Full config envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Deep-merge a fragment into the full config tree

PATCH `/config`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- property name*
  additional property — any

Responses

200 Updated config envelope (same shape as GET)

RESPONSE SCHEMA: application/json

- ok
  required — boolean
  Value: `true`
- data >
  required — object
  Unconstrained here; each operation's response narrows this via allOf.
- error
  required — object or null
  Always null on success.
- meta >
  required — object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

— **422** Semantic validation error

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 400 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Get the flattened, defaulted global settings view

GET /config/global

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Global settings envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "defaultColor": "#000000",
        "globalOutputBrightness": 1,
        "globalOutputContrast": 1,
        "globalOutputSaturation": 1,
        "globalOutputBlur": 0,
        "globalVideoAudioEnabled": true,
        "globalVideoStretchMode": 0,
        "globalVideoAudioOutputDevice": "System Default",
        "globalCaptureInputVolume": 1,
        "screenScale": 1.3,
        "language": "en",
      + "sources": { … },
        "dmxAutoConnect": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Update a subset of the global settings

PATCH `/config/global`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json

required

defaultColor	string
globalOutputBrightness	number
globalOutputContrast	number
globalOutputSaturation	number

globalOutputBlur	number
globalVideoAudioEnabled	boolean
globalVideoStretchMode	integer
globalVideoAudioOutputDevice	string
globalCaptureInputVolume	number
screenScale	number
language	string
sources >	object
sources.visualizer.devicePattern	string
sources.visualizer.fallbackPattern	string
dmxAutoConnect	boolean

Responses

200 Updated global settings envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** PATCH body has no applicable fields (code EMPTY_PATCH).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "defaultColor": "string",
    "globalOutputBrightness": 0,
    "globalOutputContrast": 0,
    "globalOutputSaturation": 0,
    "globalOutputBlur": 0,
    "globalVideoAudioEnabled": true,
    "globalVideoStretchMode": 0,
    "globalVideoAudioOutputDevice": "string",
    "globalCaptureInputVolume": 0,
    "screenScale": 0,
    "language": "string",
  - "sources": {
      + "visualizer": { … }
    },
    "sources.visualizer.devicePattern": "string",
    "sources.visualizer.fallbackPattern": "string",
    "dmxAutoConnect": true
}
```


Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "defaultColor": "#000000",
        "globalOutputBrightness": 1,
        "globalOutputContrast": 1,
        "globalOutputSaturation": 1,
        "globalOutputBlur": 0,
        "globalVideoAudioEnabled": true,
        "globalVideoStretchMode": 0,
        "globalVideoAudioOutputDevice": "System Default",
        "globalCaptureInputVolume": 1,
        "screenScale": 1.3,
        "language": "en",
      + "sources": { … },
        "dmxAutoConnect": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Layers

Visual layer visibility and per-layer configuration (elements.*).

List every valid layerId (top-level layers and addressable sub-layers)

GET /config/layer-ids

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Valid layer IDs envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Items Enum: `"imagesBackgroundView"` `"avatarsBackgroundView"` `"videoBackgroundView"` `"vinylDiscView"` `"lightingView"` `"waveFormView"` `"artistCardView"` `"marqueeTextView"` `"maskView"` `"waveFormView.waveformVisualizer"` `"waveFormView.particleWaveVisualizer"` `"waveFormView.analogWaveVisualizer"` `"waveFormView.spectrumVisualizer"` `"waveFormView.xyVisualizer"` `"lightingView.midi"` `"lightingView.particles"` `"lightingView.spotlights"` `"lightingView.strobes"` Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": [
        "imagesBackgroundView"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the raw elements object (all layers, unfiltered)

GET /config/layers

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Current layer visibility state envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Toggle visibility of a layer or sub-layer

PATCH `/config/layers/visibility`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

`layerId`
required

string (LayerId)

Enum: `"imagesBackgroundView"` `"avatarsBackgroundView"` `"videoBackgroundView"` `"vinylDiscView"` `"lightingView"` `"waveFormView"` `"artistCardView"` `"marqueeTextView"` `"maskView"` `"waveFormView.waveformVisualizer"` `"waveFormView.particleWaveVisualizer"`

`"waveFormView.analogWaveVisualizer"`
`"waveFormView.spectrumVisualizer"`
`"waveFormView.xyVisualizer"` `"lightingView.midi"`
`"lightingView.particles"` `"lightingView.spotlights"`
`"lightingView.strobes"`
Any addressable layer or sub-layer id, as returned by GET /config/layer-ids.

enabled
required

boolean

Responses

200 Current full object of the affected top-level layer (not just the changed field)

RESPONSE SCHEMA: application/json

ok
required

boolean
Value: `true`

data >
required

object
Unconstrained here; each operation's response narrows this via allOf.

error
required

object or null
Always null on success.

meta >
required

object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "layerId": "imagesBackgroundView",
    "enabled": true
}
```


Response samples

200 | 401 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get elements.<layerId>

GET /config/layers/{layerId}

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

layerId
required

string (TopLevelLayerId)
Enum: `"imagesBackgroundView"` `"avatarsBackgroundView"` `"videoBackgroundView"` `"vinylDiscView"` `"lightingView"` `"waveFormView"` `"artistCardView"` `"marqueeTextView"` `"maskView"`
One of the 9 layer ids with their own full-object GET/PATCH endpoint.

Responses

200 elements. (empty object if the section doesn't exist yet)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Deep-merge a fragment into elements.<layerId>

PATCH `/config/layers/{layerId}`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

- `layerId` — required
  string (TopLevelLayerId)
  Enum: `"imagesBackgroundView"` `"avatarsBackgroundView"` `"videoBackgroundView"` `"vinylDiscView"` `"lightingView"` `"waveFormView"` `"artistCardView"` `"marqueeTextView"` `"maskView"`
  One of the 9 layer ids with their own full-object GET/PATCH endpoint.

REQUEST BODY SCHEMA: application/json
required

non-empty

- `property name*` — additional property
  any

Responses

200 Updated elements. object

RESPONSE SCHEMA: application/json

- `ok` — required
  boolean
  Value: `true`
- `data` > — required
  object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` — required
  object or null
  Always null on success.
- `meta` > — required
  object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** PATCH body has no applicable fields (code EMPTY_PATCH).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Marquee

Individual message CRUD for the marquee text ticker carousel.

List all messages in the carousel (current order)

GET `/config/layers/marqueeTextView/messages`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Raw array of message objects

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
      + { … }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Add a new message to the carousel

POST `/config/layers/marqueeTextView/messages`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `source` required
  string (MarqueeMessageSource)
  Enum: `"custom"` `"artistName"` `"artistDescription"`
- `customText`
  string
- `enabled`
  boolean
  Default: `true`

Responses

> **201** Created message object

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy


```
{
    "source": "custom",
    "customText": "string",
    "enabled": true
}
```


Response samples

201 | 401 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "id": "string",
        "source": "custom",
        "customText": "string",
        "enabled": true
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Reorder the carousel

PATCH `/config/layers/marqueeTextView/messages/order`

Registered before the {messageId} route so this exact path is matched instead of being captured as an id.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `order` required
  Array of strings
  Exact permutation of the current message ids (same size, each id present exactly once).

Responses

200 The reordered array of full message objects

RESPONSE SCHEMA: application/json

- `ok` required
  boolean
  Value: `true`
- `data` required
  Array of objects
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required
  object or null
  Always null on success.

meta > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** 'order' isn't a valid permutation of current ids (code INVALID_ORDER).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "order": [
        "string"
    ]
}
```


Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
      + { … }
    ],
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Update one or more fields of an existing message

PATCH `/config/layers/marqueeTextView/messages/{messageId}`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

- `messageId` (required) — string

REQUEST BODY SCHEMA: application/json (required)

non-empty

- `source` — string (MarqueeMessageSource)
  Enum: `"custom"` `"artistName"` `"artistDescription"`
- `customText` — string
- `enabled` — boolean

Responses

200 The updated message object

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`

data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "source": "custom",
    "customText": "string",
    "enabled": true
}
```

Response samples

200 | 401 | 404 | 422

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
```

```
  - "data": {
        "id": "string",
        "source": "custom",
        "customText": "string",
        "enabled": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Delete a message by id

DELETE `/config/layers/marqueeTextView/messages/{messageId}`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

messageId required	string

Responses

200 Deletion confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.

error required	object or null	Always null on success.
meta > required	object (Meta)	

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "deleted": true,
        "id": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Styles

Visual style listing, the (orphaned) config.styles block, and individual style content management.

Get the raw config.styles block (orphaned, see description)

GET /config/styles

Disconnected from actual style file content. See /visuals/styles/{name} below for the real thing.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Raw styles value from config (empty object if not set)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Deep-merge a fragment into config.styles (orphaned, see description)

PATCH /config/styles

Disconnected from actual style file content. See PUT /visuals/styles/{name} below to modify a style's real content.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

property name*
additional property — any

Responses

200 Updated styles object

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 **401**

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Create a new style from the live server configuration

POST `/visuals/styles`

Captures defaultColor/currentImageFile/currentVideoFile/currentAvatarFile (resolved the same way GET /config/global resolves them) and the full current elements tree under a new name.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `name` (required) — string non-empty
  Must not contain '..', '/' or ''.
- `metadata` > — object (StyleMetadata)
- `overwrite` — boolean
  Default: `false`
- `defaultColor` — string
  Optional override; otherwise resolved from live config.
- `currentImageFile` — string
- `currentVideoFile` — string
- `currentAvatarFile` — string

Responses

> **201** Created style content

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **409** Resource with that name already exists (code ALREADY_EXISTS).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "name": "string",
  - "metadata": {
        "author": "",
        "description": "",
        "version": "1.0",
        "isInstalled": true,
        "downloadUrl": ""
    },
    "overwrite": false,
    "defaultColor": "string",
    "currentImageFile": "string",
    "currentVideoFile": "string",
    "currentAvatarFile": "string"
}
```


Response samples

201 401 409 422

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
     + "metadata": { … },
       "defaultColor": "#000000",
       "currentImageFile": "default.txt",
       "currentVideoFile": "default.txt",
       "currentAvatarFile": "default.txt",
       "elements": { }
    },
    "error": { },
  - "meta": {
       "apiVersion": "v1",
       "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
       "timestamp": "2019-08-24T14:15:22Z",
       "revision": 0
    }
}
```


Read the full content of a style

GET `/visuals/styles/{name}`

Individual Style Management -- distinct from GET /visuals/styles above (names only) and from currentStyle (GET/PATCH /config), which only tracks which name is active.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

name required	string Style name (matches a Visuals/Styles/.json file, without extension).

Responses

200 Unwrapped style content

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "metadata": { … },
        "defaultColor": "#000000",
        "currentImageFile": "default.txt",
        "currentVideoFile": "default.txt",
        "currentAvatarFile": "default.txt",
        "elements": { }
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace a style's content explicitly (creates it if missing)

PUT `/visuals/styles/{name}`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

name required	string Style name (matches a Visuals/Styles/.json file, without extension).

REQUEST BODY SCHEMA: application/json
required

metadata >	object (StyleMetadata)
defaultColor	string
currentImageFile	string
currentVideoFile	string
currentAvatarFile	string
elements >	object (ElementsTree) Raw `elements` config object (all layers, unfiltered).

Responses

200 The resulting style content

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "metadata": {
        "author": "",
        "description": "",
        "version": "1.0",
        "isInstalled": true,
        "downloadUrl": ""
    },
    "defaultColor": "string",
    "currentImageFile": "string",
    "currentVideoFile": "string",
    "currentAvatarFile": "string",
    "elements": { }
}
```

Response samples

200 | 401 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "metadata": { … },
        "defaultColor": "#000000",
        "currentImageFile": "default.txt",
        "currentVideoFile": "default.txt",
        "currentAvatarFile": "default.txt",
        "elements": { }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Deep-merge a fragment into a style's on-disk content

PATCH `/visuals/styles/{name}`

Edits Visuals/Styles/{name}.json without touching the live config -- counterpart to PUT (full replace) and POST .../apply (load onto live). MCP equivalent: patch_style(name, fragment).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

name required	string Style name (matches a Visuals/Styles/.json file, without extension).

REQUEST BODY SCHEMA: application/json
required

non-empty

metadata >	object (StyleMetadata)
defaultColor	string
currentImageFile	string
currentVideoFile	string
currentAvatarFile	string
elements >	object (ElementsTree) Raw `elements` config object (all layers, unfiltered).

Responses

200 Updated style content

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

› 422

Nothing to change (code EMPTY_PATCH) or the body has the wrong shape (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "metadata": {
        "author": "",
        "description": "",
        "version": "1.0",
        "isInstalled": true,
        "downloadUrl": ""
    },
    "defaultColor": "string",
    "currentImageFile": "string",
    "currentVideoFile": "string",
    "currentAvatarFile": "string",
    "elements": { }
}
```


Response samples

200 401 404 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
      + "metadata": { … },
        "defaultColor": "#000000",
        "currentImageFile": "default.txt",
        "currentVideoFile": "default.txt",
        "currentAvatarFile": "default.txt",
        "elements": { }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Delete a style by name

DELETE `/visuals/styles/{name}`

The "default" style is protected and cannot be deleted (checked before existence).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

name required	string Style name (matches a Visuals/Styles/.json file, without extension).

Responses

200 Deletion confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Attempted to delete the protected "default" style (code PROTECTED_STYLE).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 400 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "deleted": true,
        "name": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Atomically load a saved style onto the live config

POST `/visuals/styles/{name}/apply`

Sets currentStyle, visualization.{defaultColor,currentImageFile, currentVideoFile,currentAvatarFile}, and elements together in a single update -- unlike PATCH /config with { currentStyle: name }, which only flips the active-name pointer and relies on the desktop app to notice and reload the style from disk on its own, asynchronously. If overrides is given, it's deep-merged onto the style's own content before that atomic apply. Auto-download: if the style is only a stub (metadata.isInstalled === false with a metadata.downloadUrl set -- e.g. listed via a catalog but never fetched by this client), the package is downloaded and installed first (unzipped into Visuals/Styles, Images, Videos and Avatars, the same flow the desktop UI runs when an uninstalled style tile is clicked) before applying it. The request blocks for the duration of the download/install (up to 60s).

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

- `name` (required) — string
  Style name (matches a Visuals/Styles/.json file, without extension).

REQUEST BODY SCHEMA: application/json
optional

- `overrides` > — object (StyleOverrides)
  Deep-merged onto the style's own content before the atomic apply.

Responses

200 The values just applied to the live config

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` > (required) — object
  Unconstrained here; each operation's response narrows this via allOf.

error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **409**
Style is a stub (metadata.isInstalled === false) with no metadata.downloadUrl to fetch it from (code NOT_INSTALLED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **502**
The style needed downloading before it could be applied, and the download/install failed or timed out (60s), or the downloaded package didn't produce readable content for the requested name (code DOWNLOAD_FAILED).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "overrides": {
        "defaultColor": "string",
        "currentImageFile": "string",
        "currentVideoFile": "string",
        "currentAvatarFile": "string",
        "elements": { }
    }
}
```


Response samples

200 401 404 409 422 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "currentStyle": "string",
        "defaultColor": "string",
        "currentImageFile": "string",
        "currentVideoFile": "string",
        "currentAvatarFile": "string",
        "elements": { }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


AutoFX

Audio-reactive rule configuration (elements.audioRules).

Get elements.audioRules

GET /config/autofx

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 audioRules envelope

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Deep-merge a fragment into elements.audioRules

PATCH `/config/autofx`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

property name*
additional property — any

Responses

200 Updated audioRules object

RESPONSE SCHEMA: application/json

- ok (required) — boolean
  Value: `true`
- data (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- error (required) — object or null
  Always null on success.

meta > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Artists

Whole-list and individual artist profile management.

List all artists (normalized to an array)

GET `/config/artists`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Artists list envelope

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "artists": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace the whole artists list

PATCH /config/artists

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

artists >
required — Array of objects or object

Responses

200 Normalized artists array

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "artists": [
        { }
    ]
}
```


Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
      + "artists": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Add a single new artist

POST `/config/artists`

Granular alternative to PATCH above -- appends one artist instead of replacing the whole list.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

name required	string	non-empty
logo	string	
description	string	

Responses

201 Created artist object

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy


```
{
    "name": "string",
    "logo": "string",
    "description": "string"
}
```


Response samples

201 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "id": "string",
        "name": "string",
        "logo": "",
        "description": ""
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Reorder the artists list

PATCH `/config/artists/order`

Registered before the {artistId} route so this exact path is matched instead of being captured as an id. Adjusts artist.currentArtistIndex so the same artist (by id) stays "current" across the reorder.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `order` required
  Array of strings
  Exact permutation of the current artist ids (same size, each id present exactly once).

Responses

200 Reordered array of full artist objects

RESPONSE SCHEMA: application/json

- `ok` required
  boolean
  Value: `true`
- `data` > required
  object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required
  object or null
  Always null on success.
- `meta` > required
  object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 'order' isn't a valid permutation of current ids (code INVALID_ORDER).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "order": [
        "string"
    ]
}
```


Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "artists": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Update one or more fields of an existing artist

PATCH `/config/artists/{artistId}`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

`artistId` required	string

REQUEST BODY SCHEMA: application/json required

non-empty

`name`	string non-empty
`logo`	string
`description`	string

Responses

200 The updated artist object

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "name": "string",
    "logo": "string",
    "description": "string"
}
```

Response samples

200 | 401 | 404 | 422

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "id": "string",
        "name": "string",
        "logo": "",
        "description": ""
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Delete an artist by id

DELETE `/config/artists/{artistId}`

Adjusts artist.currentArtistIndex to keep pointing at the same logical artist.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

artistId required	string

Responses

200 Deletion confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "deleted": true,
        "id": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Playlists

Config-level image/avatar/video playlist arrays.

Get the imagePlaylists array

GET `/config/playlists/images`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 imagePlaylists value

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data`	Array of strings Absent when the value was never set.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace the imagePlaylists array wholesale

PUT /config/playlists/images

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

```
Array [
  string
]
```

Responses

200 Updated array

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta › required	object (Meta)

› **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

› **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
[
    "string"
]
```

Response samples

200 | **401** | **422**

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": [
        "string"
    ],
```

```
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the avatarPlaylists array

GET /config/playlists/avatars

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 avatarPlaylists value

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data	Array of strings Absent when the value was never set.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace the avatarPlaylists array wholesale

PUT /config/playlists/avatars

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

```
Array [
  string
]
```

Responses

200 Updated array

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
[
    "string"
]
```

Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the videoPlaylists array

GET /config/playlists/videos

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 videoPlaylists value

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data	Array of strings Absent when the value was never set.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace the videoPlaylists array wholesale

PUT /config/playlists/videos

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

```
Array [
  string
]
```

Responses

200 Updated array

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta › required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
[
    "string"
]
```

Response samples

200 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Visuals

Filesystem listings under the Visuals base folder, and local media metadata.

List style names (Visuals/Styles/*.json basenames)

GET /visuals/styles

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of style names (no extension)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta › required	object (Meta)

› **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List video playlist file names (Visuals/Videos/*.txt)

GET /visuals/playlists/videos

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of playlist names (no extension)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List image playlist file names (Visuals/Images/*.txt)

GET /visuals/playlists/images

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Array of playlist names (no extension)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List avatar playlist file names (Visuals/Avatars/*.txt)

GET /visuals/playlists/avatars

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of playlist names (no extension)

RESPONSE SCHEMA: application/json

ok
required

boolean
Value: `true`

data required	Array of strings	Unconstrained here; each operation's response narrows this via allOf.
error required	object or null	Always null on success.
meta > required	object (Meta)	

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List logo image file names (Visuals/Logos/*.{jpg,jpeg,png,gif})

GET /visuals/logos

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Array of logo file names (no extension)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data required	Array of strings Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        "string"
    ],
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List images on disk referenced by the currently active image playlist

GET /images/list

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of local media items

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
      + { … }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List videos on disk referenced by the currently active video playlist

GET /videos/list

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of local media items

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
      + { … }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

List avatars on disk referenced by the currently active avatar playlist

GET `/avatars/list`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Array of local media items

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": [
     + { … }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the styles with their install state

GET /visuals/styles-metadata

Lightweight bulk listing (name, isInstalled, downloadUrl) instead of the full style content GET /visuals/styles/{name} returns, so a client can show the install state of every style without one fetch per name.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.

error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
      + { … }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Media

External stock media (Pixabay/Pexels) search and download.

Search stock images (Pixabay/Pexels)

GET `/media/images/search`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

- `q` required — string
- `page` — integer >= 1; Default: `1`
- `perPage` — integer; Default: `0`
- `service` — string; Default: `"both"`; Enum: `"pixabay"` `"pexels"` `"both"`

Responses

200 Search results

RESPONSE SCHEMA: application/json

- `ok` required — boolean; Value: `true`
- `data` required — object; Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null; Always null on success.
- `meta` required — object (Meta)

400 Missing/invalid query parameter (code INVALID_PARAMS).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **502** External media provider error or timeout (code SEARCH_FAILED).

Response samples

200 | 400 | 401 | 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "query": "string",
     + "results": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Download a search result into Visuals/Images and the active playlist

POST /media/images/download

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

url required	string <uri>
fileName	string Must not contain '..', '/' or ''. Auto-generated from tags/timestamp when omitted.
tags	string
item >	object Optional raw provider item, forwarded to the download service.

Responses

200 Downloaded file info

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400
Body parses but has the wrong shape/required field (code INVALID_PAYLOAD, media endpoints use 400 instead of 422).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

502 External media provider download error or timeout (code DOWNLOAD_FAILED).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "url": "http://example.com",
    "fileName": "string",
    "tags": "string",
    "item": { }
}
```


Response samples

200 400 401 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "fileName": "string",
        "filePath": "string",
        "playlist": "default.txt"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Search stock videos (Pixabay/Pexels)

GET `/media/videos/search`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

- `q` — required — string
- `page` — integer `>= 1` — Default: `1`
- `perPage` — integer — Default: `0`
- `service` — string — Default: `"both"` — Enum: `"pixabay"` `"pexels"` `"both"`

Responses

200 Search results

RESPONSE SCHEMA: application/json

- `ok` — required — boolean — Value: `true`
- `data` — required — object — Unconstrained here; each operation's response narrows this via allOf.
- `error` — required — object or null — Always null on success.
- `meta` — required — object (Meta)

400 Missing/invalid query parameter (code INVALID_PARAMS).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **502** External media provider error or timeout (code SEARCH_FAILED).

Response samples

200 | 400 | 401 | 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "query": "string",
      + "results": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Download a search result into Visuals/Videos and the active playlist

POST /media/videos/download

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

url required	string <uri>
fileName	string Must not contain '..', '/' or ''. Auto-generated from tags/timestamp when omitted.
tags	string
item >	object Optional raw provider item, forwarded to the download service.

Responses

200 Downloaded file info

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400
Body parses but has the wrong shape/required field (code INVALID_PAYLOAD, media endpoints use 400 instead of 422).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

502 External media provider download error or timeout (code DOWNLOAD_FAILED).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "url": "http://example.com",
    "fileName": "string",
    "tags": "string",
    "item": { }
}
```


Response samples

200 400 401 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "fileName": "string",
        "filePath": "string",
        "playlist": "default.txt"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Search stock avatars (reuses the image search -- an avatar is just an image saved to a different folder/playlist)

GET /media/avatars/search

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

- `q` required — string
- `page` — integer `>= 1`; Default: `1`
- `perPage` — integer; Default: `0`
- `service` — string; Default: `"both"`; Enum: `"pixabay"` `"pexels"` `"both"`

Responses

200 Search results

RESPONSE SCHEMA: application/json

- `ok` required — boolean; Value: `true`
- `data` > required — object; Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null; Always null on success.
- `meta` > required — object (Meta)

400 Missing/invalid query parameter (code INVALID_PARAMS).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

502 External media provider error or timeout (code SEARCH_FAILED).

Response samples

200 | 400 | 401 | 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "query": "string",
      + "results": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Download a search result into Visuals/Avatars and the active playlist

POST /media/avatars/download

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

url
required — string <uri>

- `fileName` — string
  Must not contain '..', '/' or ''. Auto-generated from tags/timestamp when omitted.
- `tags` — string
- `item` > — object
  Optional raw provider item, forwarded to the download service.

Responses

200 Downloaded file info

RESPONSE SCHEMA: application/json

- `ok` *required* — boolean
  Value: `true`
- `data` > *required* — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` *required* — object or null
  Always null on success.
- `meta` > *required* — object (Meta)

400
Body parses but has the wrong shape/required field (code INVALID_PAYLOAD, media endpoints use 400 instead of 422).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

502 External media provider download error or timeout (code DOWNLOAD_FAILED).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "url": "http://example.com",
    "fileName": "string",
    "tags": "string",
    "item": { }
}
```


Response samples

200 400 401 502

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "fileName": "string",
        "filePath": "string",
        "playlist": "default.txt"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


DMX

DMX fixtures, scenes, movement/laser patterns, fixture profiles, runtime control, and export/import.

Get all DMX fixtures

GET `/dmx/fixtures`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Fixtures array

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	Array of objects Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

500 Local read/write failure (code INTERNAL_ERROR).

Response samples

200 | 401 | 500

Content type
application/json

```
Copy  Expand all  Collapse all
{
    "ok": true,
  - "data": [
        { }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace all DMX fixtures

PUT `/dmx/fixtures`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

One of **DmxFixture** | DmxFixturesWrapper

```
Array [
  property name*
  additional property   any
]
```

Responses

200 Saved fixtures array

RESPONSE SCHEMA: application/json

ok
required

boolean
Value: `true`

data > required	Array of objects	Unconstrained here; each operation's response narrows this via allOf.
error required	object or null	Always null on success.
meta > required	object (Meta)	

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** DMX fixtures/scenes failed schema validation (code VALIDATION_ERROR).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Example
DmxFixture

Copy

```
[ ]
```

Response samples

200 | 401 | 422 | 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": [
        { }
    ],
```

```
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace all DMX fixtures (object body only, unlike PUT)

PATCH `/dmx/fixtures`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `fixtures` > required — Array of objects (DmxFixture)

Responses

200 Saved fixtures array

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` > required — Array of objects
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "fixtures": [
        { }
    ]
}
```


Response samples

200 401 422 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": [
        { }
    ],
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
```

```
}
```

Get the full contents of dmx_scenes.json

GET `/dmx/scenes`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Scenes root object

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Replace the full scenes root object

PUT /dmx/scenes

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

One of **DmxScenesRoot** Array of objects

property name*
additional property — any

Responses

200 Saved scenes root object

RESPONSE SCHEMA: application/json

ok
required — boolean
Value: `true`

data >	required	object Unconstrained here; each operation's response narrows this via allOf.
error	required	object or null Always null on success.
meta >	required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** DMX fixtures/scenes failed schema validation (code VALIDATION_ERROR).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Example
DmxScenesRoot

Copy

```
{ }
```

Response samples

200 | 401 | 422 | 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
    "data": { },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Shallow-merge into the scenes root object

PATCH `/dmx/scenes`

If the body includes a `scenes` key, the entire scenes array is replaced, not appended to -- there is no per-scene merge.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

property name*
additional property — any

Responses

200 Updated scenes root object

RESPONSE SCHEMA: application/json

- ok (required) — boolean
  Value: `true`
- data > (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- error (required) — object or null
  Always null on success.

meta > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 401 | 422 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Get the raw fixture movement pattern file

GET `/dmx/movement/fixtures/patterns`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Raw pattern file contents

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Overwrite the fixture movement pattern file (no schema validation)

PUT /dmx/movement/fixtures/patterns

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

property name*
additional property — any

Responses

200 The body echoed back

RESPONSE SCHEMA: application/json

ok
required — boolean
Value: `true`

data >
required — object
Unconstrained here; each operation's response narrows this via allOf.

error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 401 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Get the raw laser movement pattern file

GET `/dmx/movement/laser/patterns`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Raw pattern file contents

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`	
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.	
`error` required	object or null Always null on success.	
`meta` > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Copy Expand all Collapse all

Overwrite the laser movement pattern file (no schema validation)

PUT /dmx/movement/laser/patterns

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

property name*
additional property — any

Responses

200 The body echoed back

RESPONSE SCHEMA: application/json

ok
required — boolean
Value: `true`

data >
required — object
Unconstrained here; each operation's response narrows this via allOf.

error
required

object or null

Always null on success.

meta >
required

object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 401 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
    "data": { },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Get DMX fixture color/behavior profiles

GET `/dmx/fixtures/profiles`

If the DMX runtime helper isn't available, responds 200 with { "profiles": [] } rather than an error.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Profiles list

RESPONSE SCHEMA: application/json

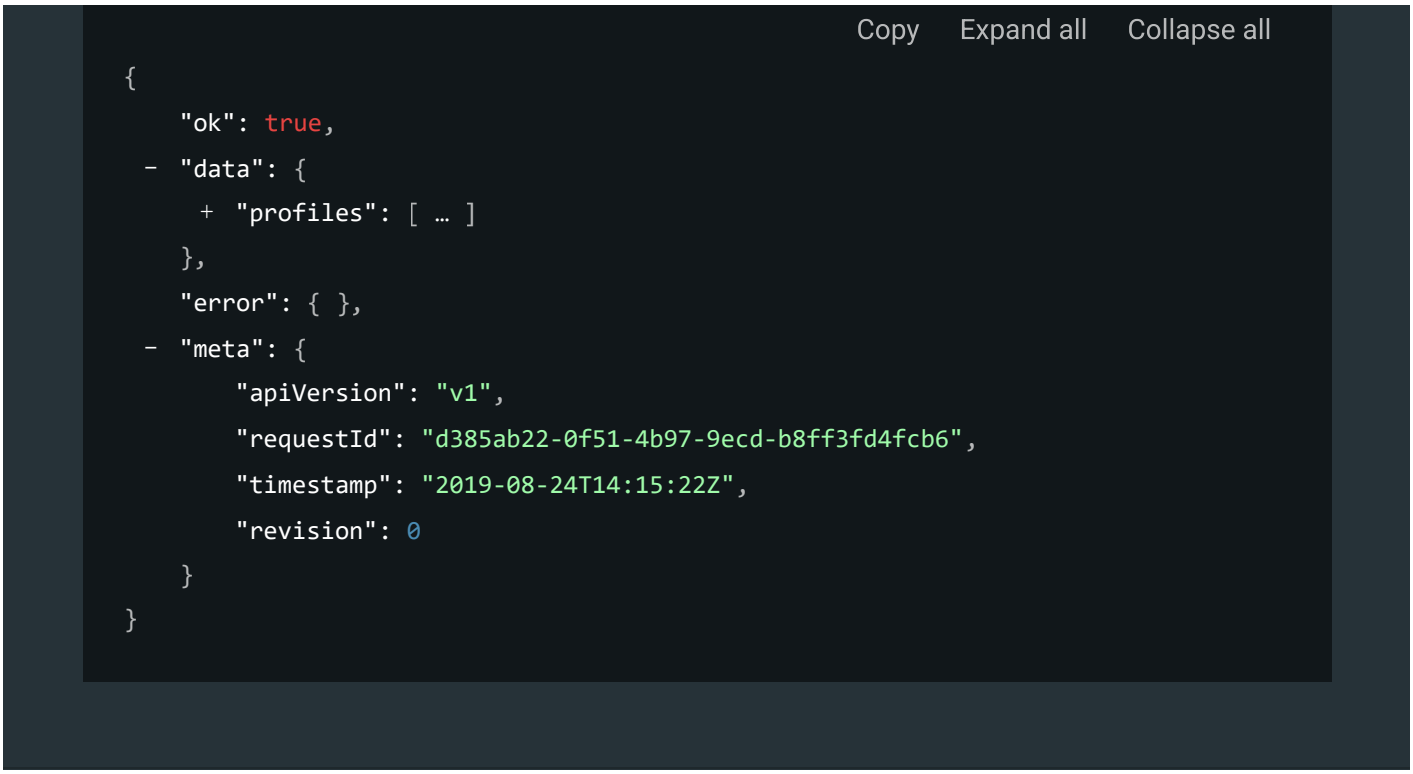
`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json



Replace all DMX fixture profiles

PUT /dmx/fixtures/profiles

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json

required

One of | Array of objects | DmxFixtureProfilesData

Array [

property name* — any

additional property

]

Responses

200 Reloaded profiles list

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **500** Local read/write failure (code INTERNAL_ERROR).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Example

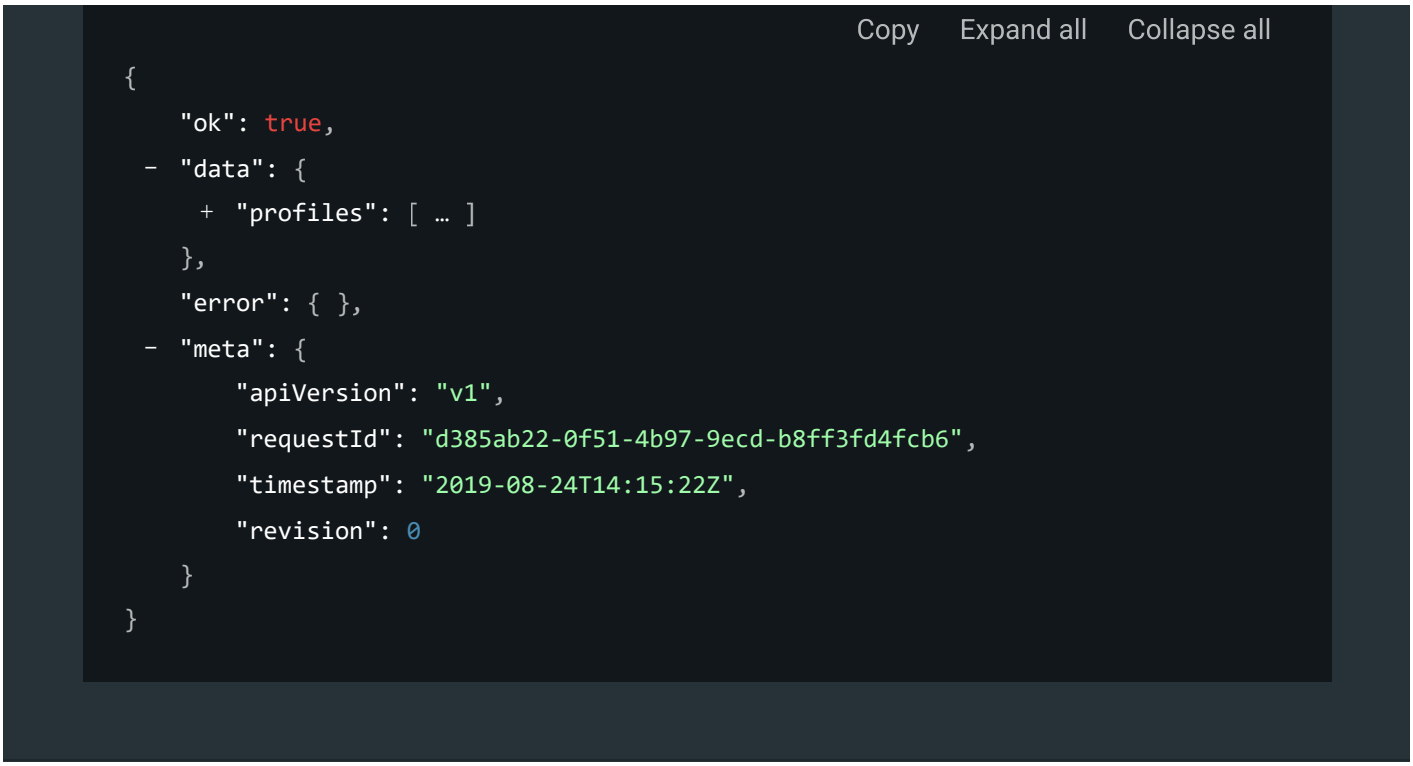
Copy

```
[ ]
```

Response samples

200 | 401 | 422 | 500 | 503

Content type
application/json



Replace all DMX fixture profiles (full replace, same as PUT)

PATCH /dmx/fixtures/profiles

Patch behaves as replace for now (same as PUT), to keep parity with list semantics.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

One of | Array of objects | DmxFixtureProfilesData

```
Array [
  property name*
  additional property   any
]
```

Responses

200 Reloaded profiles list

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **500** Local read/write failure (code INTERNAL_ERROR).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Example

Copy

```
[ ]
```

Response samples

200 401 422 500 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "profiles": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Instantly (or progressively) apply a DMX scene

POST `/dmx/runtime/scene/apply`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- **scene** > — object
  Full scene object, applied directly.
- **sceneId** — string
  Looked up in dmx_scenes.json.
- **progress** — number [0 .. 1]
  If present and >= 0, applies with progress instead of instantly.

Responses

200 Applied confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "scene": { },
    "sceneId": "string",
    "progress": 1
}
```

Response samples

200 401 404 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "applied": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Apply raw DMX channel values immediately

PATCH `/dmx/runtime/channels`

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `universe` — integer
  Default: 0
- `channels` > (required) — Array of objects

Responses

200 Number of channels processed

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "universe": 0,
  - "channels": [
      + { … }
    ]
}
```


Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "updated": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Toggle global DMX blackout

POST /dmx/runtime/blackout

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

active — boolean
Default: `true`

Responses

200 Resulting blackout state

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy

```
{
    "active": true
}
```

Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
```

```
  - "data": {
        "active": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Audio-silence state of the DMX per-element silence behavior

GET /dmx/runtime/silence

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Silence state

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data > required	object Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "active": true,
        "fadeMs": 5000,
        "curve": "linear"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Force the audio-silence state and/or set its global fade

POST /dmx/runtime/silence

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json

required

active	boolean
fadeMs	integer [0 .. 5000]
curve	string Enum: `"linear"` `"smooth"` `"easeIn"` `"easeOut"`

Responses

200 Silence state

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Neither field present, or a value out of range

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy


```
{
    "active": true,
    "fadeMs": 5000,
    "curve": "linear"
}
```


Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "active": true,
        "fadeMs": 5000,
        "curve": "linear"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Control global movement speed and enabled state

PATCH /dmx/runtime/movement

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json

required

enabled	boolean
globalSpeed	number

Responses

200 OK confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type

application/json

Copy

```
{
    "enabled": true,
    "globalSpeed": 0
}
```


Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "ok": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Toggle auto-dimmer mode and/or set its level

PATCH /dmx/runtime/dimmer

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `enabled` — boolean
  Toggles auto-dimmer mode.
- `autoValue` — integer [0 .. 255]

Level used while auto-dimmer mode is on.

Responses

200 OK confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy

```
{
    "enabled": true,
    "autoValue": 255
}
```

Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "ok": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Trigger a DMX macro (fog, haze, laser-blackout, strobe)

POST /dmx/runtime/macros/{macro}

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

macro
required
string (DmxMacroName)
Enum: "fog" "haze" "laser-blackout" "strobe"

REQUEST BODY SCHEMA: application/json
optional

intensity
number
Default: 1
No range validation applied.

durationMs
integer

fog only. Burst duration in ms, server-clamped to [50, 30000]. Omit (or send <= 0) for the default short burst (~900ms); pass several thousand ms for a long burst.

color

string

strobe only. Hex flash color, e.g. #FFFFFF. Invalid/empty defaults to white.

Responses

200 Triggered confirmation

RESPONSE SCHEMA: application/json

ok
required

boolean

Value: `true`

data >
required

object

Unconstrained here; each operation's response narrows this via allOf.

error
required

object or null

Always null on success.

meta >
required

object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Unknown DMX macro name (code INVALID_MACRO, 404 rather than 400/422).

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type

application/json

Copy

```
{
    "intensity": 1,
    "durationMs": 0,
    "color": "string"
}
```

Response samples

200 | 401 | 404 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "triggered": "fog"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Bundle the current DMX state into one object

GET /dmx/export

Always succeeds -- missing files degrade to empty defaults, not errors.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Full DMX snapshot

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "fixtures": [ … ],
        "scenes": { },
        "movementPatterns": { },
        "laserPatterns": { },
      + "fixtureProfiles": { … }
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Import a DMX state snapshot

POST /dmx/import

An entirely empty body {} is accepted and is effectively a no-op success. See DmxImportRequest for the merge-vs-replace and scenes-shape caveats.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

- **mode** — string
  'merge' merges fixtures/scenes by id (incoming wins); anything else (including omitted) is a full replace. movementPatterns/laserPatterns/fixtureProfiles are always a full replace regardless of mode.
- **dryRun** — boolean
  Default: `false`
  When true, validation still runs but nothing is written or reloaded.
- **fixtures** > — Array of objects (DmxFixture)
- **scenes** > — Array of objects
- **movementPatterns** > — object (DmxPatternFile)
  Raw contents of a movement/laser pattern file. No schema validation is applied -- written/read as-is.
- **laserPatterns** > — object (DmxPatternFile)
  Raw contents of a movement/laser pattern file. No schema validation is applied -- written/read as-is.

fixtureProfiles > Array of objects

Responses

200 Import confirmation

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 DMX fixtures/scenes failed schema validation (code VALIDATION_ERROR).

500 Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "mode": "merge",
    "dryRun": false,
```

```
  - "fixtures": [
        { }
    ],
  - "scenes": [
        { }
    ],
    "movementPatterns": { },
    "laserPatterns": { },
  - "fixtureProfiles": [
        { }
    ]
}
```


Response samples

200 | 401 | 422 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "imported": true,
        "mode": "string",
        "dryRun": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Broadcast the Play All state to the other clients

POST /dmx/runtime/playall

Pure coordination: nothing is stored on the server. It fans running out to every connected client (WebSocket event dmx.playall.changed), so a client that connects later sees false until the next change.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- running — boolean
  Default: `false`

Responses

200 OK

RESPONSE SCHEMA: application/json

- ok (required) — boolean
  Value: `true`
- data > (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- error (required) — object or null
  Always null on success.
- meta > (required) — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Request samples

Payload

Content type
application/json

Copy


```
{
    "running": true
}
```


Response samples

200 400 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "running": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Force a scene playback mode on every client

POST /dmx/runtime/scene-mode

Pure coordination, same as /dmx/runtime/playall (WebSocket event dmx.scenemode.changed). Any value other than auto, manual or mixto is stored as perScene (each scene plays its own saved mode).

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- mode
  string
  Enum: `"auto"` `"manual"` `"mixto"` `"perScene"`
  Any other value is answered as perScene.

Responses

200 OK

RESPONSE SCHEMA: application/json

- ok
  required
  boolean
  Value: `true`
- data >
  required
  object
  Unconstrained here; each operation's response narrows this via allOf.
- error
  required
  object or null
  Always null on success.
- meta >
  required
  object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Request samples

Payload

Content type
application/json

Copy

```
{
    "mode": "auto"
}
```


Response samples

200 | 400 | 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "mode": "auto"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the live DMX runtime state

GET /dmx/runtime/status

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "connected": true,
        "connectedPortName": "COM5",
        "connectedUniverseCount": 1,
        "packetCount": 18342,
        "blackoutActive": false,
        "movementEngineEnabled": true,
        "movementGlobalSpeed": 1,
        "dimmerAutoMode": false,
        "autoDimmerValue": 255,
        "fixtureStrobesEnabled": true,
        "audioSilent": false,
        "silenceFadeMs": 400,
        "silenceFadeCurve": "smooth"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Read a fixture's channels by role

GET `/dmx/runtime/fixtures/{fixtureId}/channels`

Addresses channels as the absolute DMX channels the fixture occupies (the way the desktop's per-fixture fine-control panel does), so a client does not have to duplicate the fixture to port/universe routing. Only channels inside 1-512 are listed.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

`fixtureId` required	string Fixture id, as in GET /dmx/fixtures.

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

500 Local read/write failure (code INTERNAL_ERROR).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 404 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "fixtureId": "par-1",
      + "channels": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Set a fixture's channels

PATCH `/dmx/runtime/fixtures/{fixtureId}/channels`

Same addressing as the GET. Broadcasts dmx.runtime.changed.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

`fixtureId` required	string Fixture id, as in GET /dmx/fixtures.

REQUEST BODY SCHEMA: application/json required

`channels` > required	Array of objects

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
> The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "channels": [
      + { … }
    ]
}
```


Response samples

200 400 401 503

Content type
application/json

```
Copy  Expand all  Collapse all
{
    "ok": true,
  - "data": {
        "fixtureId": "string",
        "updated": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Recording

Start/stop/status/list/download of MP4 recordings of the main application window, plus single-frame PNG snapshots.

Start recording the main application window to an MP4 file

POST `/recording/start`

Only one recording can be active at a time. maxDurationSec enforces an auto-stop safety net since this can be triggered remotely with no on-screen recording indicator. Audio is recorded automatically from whichever input device is currently selected for audio analysis (falling back to the system default input); there is no separate toggle for it.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

fps
integer
Default: `20`
Clamped to [5, 60].

maxDurationSec
integer
Default: `1800`
Clamped to [5, 7200]; the recording auto-stops when reached.

fileName
string
Sanitized against path traversal; .mp4 appended if missing; auto-generated timestamped name if omitted.

Responses

200 Recording started

RESPONSE SCHEMA: application/json

ok
required
boolean
Value: `true`

data >
required
object
Unconstrained here; each operation's response narrows this via allOf.

error
required
object or null
Always null on success.

meta >
required
object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 A screen recording is already in progress (code ALREADY_RECORDING).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy


```
{
    "fps": 20,
    "maxDurationSec": 1800,
    "fileName": "string"
}
```


Response samples

200 | 400 | 401 | 409 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "recording": true,
        "fileName": "string",
        "fps": 0,
        "maxDurationSec": 0,
        "hasAudio": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Stop and finalize the current recording

POST `/recording/stop`

Blocks briefly server-side (up to ~5s) until the encoder confirms the file is fully written before responding.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Recording stopped

RESPONSE SCHEMA: application/json

	required	
ok	required	boolean Value: `true`
data	required	object Unconstrained here; each operation's response narrows this via allOf.
error	required	object or null Always null on success.
meta	required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 No screen recording is currently in progress (code NOT_RECORDING).

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 401 409 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "recording": false,
        "fileName": "string",
        "sizeBytes": 0,
        "durationSec": 0,
        "hasAudio": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the current recording status

GET `/recording/status`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Recording status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "recording": true,
        "fileName": "string",
        "elapsedSec": 0,
        "fps": 0,
        "hasAudio": true
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List saved recordings

GET /recording/list

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Recordings list, newest first

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

› 503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "recordings": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Download a saved recording

GET `/recording/download/{fileName}`

Returns the raw MP4 bytes, not the standard ok/data envelope -- fileName is sanitized against path traversal.

AUTHORIZATIONS: › *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

fileName
required

string

Responses

200 MP4 video bytes

RESPONSE SCHEMA: video/mp4

string <binary>

400 Invalid fileName (path traversal attempt)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": false,
    "data": { },
  - "error": {
        "code": "string",
        "message": "string",
        "hint": "string",
        "details": { }
    },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Capture a single PNG snapshot of the main application window

POST `/recording/snapshot`

Independent of any in-progress video recording -- can be called whether or not /recording/start is active.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

- `fileName` — string
  Sanitized against path traversal; .png appended if missing; auto-generated timestamped name if omitted.

Responses

200 Snapshot captured

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` — object or null

required	Always null on success.
meta > required	object (Meta)

— 400 Snapshot failed (e.g. main window not visible yet)

> 401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> 503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy

```
{
    "fileName": "string"
}
```

Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "fileName": "string",
        "width": 0,
        "height": 0,
        "sizeBytes": 0
    },
```

```
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List saved snapshots

GET `/recording/snapshots`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Snapshots list, newest first

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "snapshots": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Download a saved snapshot

GET `/recording/snapshot/download/{fileName}`

Returns the raw PNG bytes, not the standard ok/data envelope -- fileName is sanitized against path traversal.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

fileName
required

string

Responses

200 PNG image bytes

RESPONSE SCHEMA: image/png

string <binary>

400 Invalid fileName (path traversal attempt)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": false,
    "data": { },
  - "error": {
        "code": "string",
        "message": "string",
        "hint": "string",
        "details": { }
    },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Spout

Spout output (Windows only, builds with the Spout SDK): share the visual output as a texture with OBS, Resolume, TouchDesigner, VJ apps... on the same computer, with no encoding. Status, settings, start/stop and the list of senders registered on the machine.

Get the live state of the Spout output

GET `/spout/status`

Always answers, also on a build without the SDK (available: false). A stopped output has null for everything that comes from a frame.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Spout status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.

error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "running": true,
        "stalled": true,
        "path": "gpu",
        "degraded": true,
        "degradedReason": "string",
        "renderingApi": "direct3d11",
        "senderName": "string",
        "effectiveSource": "output",
        "effectiveFps": 0,
      + "outputSize": { … },
        "framesSent": 0,
        "lastFrameAgeMs": 0,
        "adapterName": "string",
        "lastError": "string",
      + "config": { … }
    },
```

```
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the Spout output settings

GET /spout/config

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Spout settings

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

› 503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "senderName": "string",
        "fps": 120,
        "scale": 0.1,
        "width": 0,
        "height": 0,
        "source": "output",
        "mode": "auto",
        "adapterIndex": -1,
      + "adapters": [ … ],
        "autoStart": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Change some of the Spout output settings

PATCH `/spout/config`

Persists the change (the same store as the Settings > Spout Output panel). Works while the output is stopped or running: see SpoutConfigPatch for what applies live and what recreates the sender. The answer is the resulting settings.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

`senderName`	string
`fps`	integer
`scale`	number
`width`	integer
`height`	integer
`source`	string or integer
`mode`	string or integer
`adapterIndex`	integer
`autoStart`	boolean

Responses

200 Settings after the change

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`

data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422

Not a JSON object, unknown field, wrong type, value out of range (code INVALID_PAYLOAD; the message lists every problem) or nothing to change (code EMPTY_PATCH).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Request samples

Payload

Content type
application/json

Copy

```
{
    "senderName": "string",
    "fps": 0,
    "scale": 0,
    "width": 0,
    "height": 0,
    "source": "output",
    "mode": "auto",
    "adapterIndex": 0,
    "autoStart": true
}
```

Response samples

200 | 400 | 401 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "senderName": "string",
        "fps": 120,
        "scale": 0.1,
        "width": 0,
        "height": 0,
        "source": "output",
        "mode": "auto",
        "adapterIndex": -1,
      + "adapters": [ … ],
        "autoStart": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Start the Spout output

POST /spout/start

The body is optional. Sending settings starts with them and stores them only if the output really starts. The GPU path needs the application window to be shown: a hidden or minimised window

renders nothing, so the output would report stalled. A receiver must run on the same GPU as the sender (with several GPUs use mode cpu and adapterIndex).

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

senderName	string
fps	integer
scale	number
width	integer
height	integer
source >	string or integer
mode >	string or integer
adapterIndex	integer
autoStart	boolean

Responses

200 Output started; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
The active edition does not include this feature (code EDITION_REQUIRED; the message names the edition). See x-edition on the operation.

> **409**
The Spout output is already running (code ALREADY_RUNNING). The running output is not touched and a start body is not applied.

> **422** Not a JSON object, unknown field, wrong type or value out of range (code INVALID_PAYLOAD).

> **500**
The output could not start (code SPOUT_START_FAILED; the message says why: no window yet, no compatible GPU, Spout refused the sender...). Nothing is changed, not even the settings sent in the body.

> **503**
The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Request samples

Payload

Content type
application/json

Copy

```
{
    "senderName": "string",
    "fps": 0,
    "scale": 0,
    "width": 0,
    "height": 0,
    "source": "output",
    "mode": "auto",
    "adapterIndex": 0,
    "autoStart": true
```

```
}
```

Response samples

200 | 400 | 401 | 403 | 409 | 422 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "running": true,
        "stalled": true,
        "path": "gpu",
        "degraded": true,
        "degradedReason": "string",
        "renderingApi": "direct3d11",
        "senderName": "string",
        "effectiveSource": "output",
        "effectiveFps": 0,
      + "outputSize": { … },
        "framesSent": 0,
        "lastFrameAgeMs": 0,
        "adapterName": "string",
        "lastError": "string",
      + "config": { … }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Stop the Spout output

POST `/spout/stop`

Removes the sender from Spout. The settings are kept.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Output stopped; the body is the status

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`	
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.	
`error` required	object or null Always null on success.	
`meta` > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 The Spout output is not running (code NOT_RUNNING).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 401 409 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "running": true,
        "stalled": true,
        "path": "gpu",
        "degraded": true,
        "degradedReason": "string",
        "renderingApi": "direct3d11",
        "senderName": "string",
        "effectiveSource": "output",
        "effectiveFps": 0,
      + "outputSize": { … },
        "framesSent": 0,
        "lastFrameAgeMs": 0,
        "adapterName": "string",
        "lastError": "string",
      + "config": { … }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the Spout senders registered on this machine

GET /spout/senders

To check that a receiver can see this application (its own sender is listed when running).

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Sender names

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
      + "senders": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Read a Spout layout file

GET `/spout/layout`

Tolerant read (an output with unusable data is dropped or falls back; it never rejects the file). valid/validationErrors report the strict check PUT would apply to this same file. A file that is not a JSON object at all is 422 INVALID_PAYLOAD.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

`path` — string
Layout file, relative to the application folder or absolute. Default spout_config/spout_layout_config.example.json.

Responses

200 OK

RESPONSE SCHEMA: application/json

`ok` required — boolean
Value: `true`

`data` — object

required	Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **503**
The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 404 | 422 | 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "path": "C:/DjMirrorPulse/spout_config/my_layout.json",
        "valid": true,
        "warnings": [ ],
      + "outputs": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Write a Spout layout file

PUT `/spout/layout`

The whole layout is validated strictly (unique non-empty names, at most 16 outputs, a valid shape per mask type...) and refused as one if anything is wrong: no partial save. Written exactly as sent; if path is the running layout it is reloaded at once. Broadcasts spout.layout.updated.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

path
string
Layout file, relative to the application folder or absolute. Default spout_config/spout_layout_config.example.json.

REQUEST BODY SCHEMA: application/json
required

outputs >
required
Array of objects (LayoutOutput) <= 16 items

Responses

200 OK

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

500 Local read/write failure (code INTERNAL_ERROR).

503
The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
  - "outputs": [
      + { … }
    ]
}
```


Response samples

200 | 400 | 401 | 422 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "path": "C:/DjMirrorPulse/spout_config/my_layout.json",
        "valid": true,
        "warnings": [ ],
      + "outputs": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the live state of the Spout layout service

GET /spout/layout/status

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "running": true,
        "path": "gpu",
        "degraded": false,
        "layoutPath": "C:/DjMirrorPulse/spout_config/my_layout.json",
      + "textureSize": { … },
        "outputCount": 1,
        "activeSenderCount": 1,
      + "outputs": [ … ],
        "lastError": null
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the Spout layout service settings

GET `/spout/layout/config`

The service's own settings, not the layout file's content.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

`ok` required — boolean

Value: `true`

data >	required	object Unconstrained here; each operation's response narrows this via allOf.
error	required	object or null Always null on success.
meta >	required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "layoutPath": "string",
        "mode": "auto",
        "source": "output",
        "fps": 120,
        "adapterIndex": -1,
      + "adapters": [ … ],
        "autoStart": true
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change the Spout layout service settings

PATCH `/spout/layout/config`

Unlike PATCH /spout/config every field applies immediately (a layoutPath change reloads a running layout in place). An empty patch is 422 EMPTY_PATCH. Broadcasts spout.layout.updated.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `layoutPath` — string
- `mode` — string
  Enum: `"auto"` `"gpu"` `"cpu"`
- `source` — string
  Enum: `"output"` `"window"`
- `fps` — integer `[ 0 .. 120 ]`
- `adapterIndex` — integer `>= -1`
- `autoStart` — boolean

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **503**
The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Request samples

Payload

Content type
application/json

Copy

```
{
    "fps": 30,
    "autoStart": true
}
```

Response samples

200 | 400 | 401 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "layoutPath": "string",
        "mode": "auto",
        "source": "output",
        "fps": 120,
        "adapterIndex": -1,
      + "adapters": [ … ],
        "autoStart": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Start the Spout layout service

POST /spout/layout/start

The body is optional, with the same fields as PATCH /spout/layout/config; they apply immediately even if the start then fails. Broadcasts spout.layout.started.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

layoutPath string

mode	string Enum: `"auto"` `"gpu"` `"cpu"`
source	string Enum: `"output"` `"window"`
fps	integer `[ 0 .. 120 ]`
adapterIndex	integer `>= -1`
autoStart	boolean

Responses

200 Started; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403

The active edition does not include this feature (code EDITION_REQUIRED; the message names the edition). See x-edition on the operation.

409

The Spout output is already running (code ALREADY_RUNNING). The running output is not touched and a start body is not applied.

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> 500

The output could not start (code SPOUT_START_FAILED; the message says why: no window yet, no compatible GPU, Spout refused the sender...). Nothing is changed, not even the settings sent in the body.

> 503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Request samples

Payload

Content type
application/json

Copy

```
{
    "fps": 30,
    "autoStart": true
}
```

Response samples

200 | 400 | 401 | 403 | 409 | 422 | 500 | 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "running": true,
        "path": "gpu",
        "degraded": false,
        "layoutPath": "C:/DjMirrorPulse/spout_config/my_layout.json",
      + "textureSize": { … },
        "outputCount": 1,
        "activeSenderCount": 1,
      + "outputs": [ … ],
        "lastError": null
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Stop the Spout layout service

POST `/spout/layout/stop`

Broadcasts spout.layout.stopped.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Stopped; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`

data > required	object	Unconstrained here; each operation's response narrows this via allOf.
error required	object or null	Always null on success.
meta > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 The Spout output is not running (code NOT_RUNNING).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 409 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "running": true,
        "path": "gpu",
        "degraded": false,
        "layoutPath": "C:/DjMirrorPulse/spout_config/my_layout.json",
      + "textureSize": { … },
        "outputCount": 1,
        "activeSenderCount": 1,
      + "outputs": [ … ],
        "lastError": null
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Re-read the layout file now

POST /spout/layout/reload

Instead of waiting for the 500 ms watch. If the file went missing or stopped parsing the previous layout keeps running.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Reloaded; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

409 The Spout output is not running (code NOT_RUNNING).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

503
The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 | 401 | 404 | 409 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "running": true,
        "path": "gpu",
        "degraded": false,
        "layoutPath": "C:/DjMirrorPulse/spout_config/my_layout.json",
      + "textureSize": { … },
        "outputCount": 1,
        "activeSenderCount": 1,
      + "outputs": [ … ],
        "lastError": null
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

List the saved layout files

GET /spout/layout/files

Base names (no extension) of the .json files under spout_config/, to build a ?path= picker. Empty, not an error, before the folder has anything in it.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

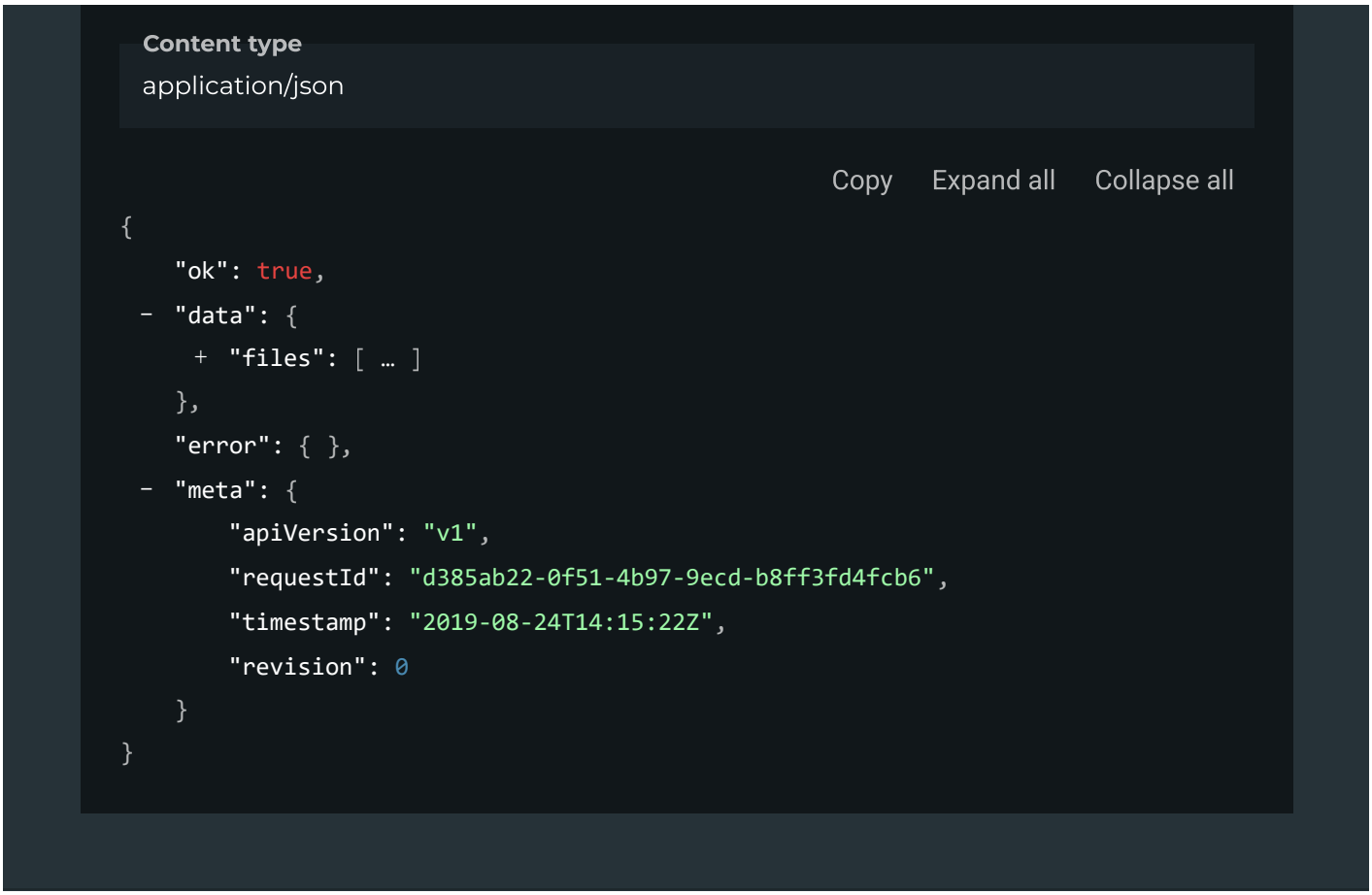
401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The Spout output cannot start on this build/OS (code NOT_AVAILABLE; the message says why: Windows only, needs the Spout SDK), or the service is not wired into the server.

Response samples

200 401 503



Link

Ableton Link (desktop builds with the Link SDK): shares tempo and bar phase over the LAN with Traktor, Ableton Live, Resolume and other Link apps. DjMirrorPulse only follows the session, it never changes the session tempo. Live state and settings.

Get the live Ableton Link session state

GET `/link/status`

Read-only. Also answers on a build without the SDK (available false, everything else 0 / false).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Link session state

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503 The Ableton Link service is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "enabled": true,
        "connected": true,
        "peerCount": 0,
        "bpm": 0,
        "phase": 0,
        "beatInBar": 0,
        "quantum": 0,
        "phaseOffsetMs": 0,
        "barOffsetBeats": 31,
        "clockSource": 0,
        "clockSourceName": "audio",
        "beatPolicy": 0,
        "beatPolicyName": "linkGrid",
        "fallbackActive": true,
        "bpmSource": "link",
        "effectiveBpm": 0,
        "otherSessionSeen": true,
        "rejoinAttempts": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the Ableton Link settings

GET /link/config

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Link settings

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503** The Ableton Link service is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "enabled": true,
        "quantum": 1,
        "phaseOffsetMs": -500,
        "barOffsetBeats": 31,
        "clockSource": 0,
        "clockSourceName": "audio",
        "beatPolicy": 0,
        "beatPolicyName": "linkGrid",
        "mirrorLinkBeatsToLegacySignal": true,
        "beatWindowMs": 10,
        "tempoSmoothingMs": 2000,
        "fallbackToAudioOnDisconnect": true,
        "beatsDuringSilence": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change some of the Ableton Link settings

PATCH `/link/config`

Persists the change (the same store as the Settings > Ableton Link panel) and applies it live. The answer is the resulting settings.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

enabled — boolean

quantum	integer [1 .. 32]
phaseOffsetMs	integer [-500 .. 500]
barOffsetBeats	integer [0 .. 31]
clockSource ›	string or integer
beatPolicy ›	string or integer
mirrorLinkBeatsToLegacySignal	boolean
beatWindowMs	integer [10 .. 200]
tempoSmoothingMs	integer [0 .. 2000]
fallbackToAudioOnDisconnect	boolean
beatsDuringSilence	boolean

Responses

200 Settings after the change

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data › required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta › required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403
The active edition does not include this feature (code EDITION_REQUIRED; the message names the edition). See x-edition on the operation.

422
Not a JSON object, unknown field, wrong type or value out of range (code INVALID_PAYLOAD; the message names the field), nothing to change (code EMPTY_PATCH), or enabled true on a build without the Link SDK (code LINK_UNAVAILABLE).

503 The Ableton Link service is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "enabled": true,
    "quantum": 1,
    "phaseOffsetMs": -500,
    "barOffsetBeats": 31,
    "clockSource": "audio",
    "beatPolicy": "linkGrid",
    "mirrorLinkBeatsToLegacySignal": true,
    "beatWindowMs": 10,
    "tempoSmoothingMs": 2000,
    "fallbackToAudioOnDisconnect": true,
    "beatsDuringSilence": true
}
```

Response samples

200 400 401 403 422 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "enabled": true,
        "quantum": 1,
        "phaseOffsetMs": -500,
        "barOffsetBeats": 31,
        "clockSource": 0,
        "clockSourceName": "audio",
        "beatPolicy": 0,
        "beatPolicyName": "linkGrid",
        "mirrorLinkBeatsToLegacySignal": true,
        "beatWindowMs": 10,
        "tempoSmoothingMs": 2000,
        "fallbackToAudioOnDisconnect": true,
        "beatsDuringSilence": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Make the Link beat playing now the 1 of the bar

POST `/link/downbeat`

Call it when you hear the first beat of the bar (a little before or after is fine): the Link beat nearest to now becomes beat 1, which sets barOffsetBeats. It persists and applies live, and only shifts this application's own bar count: the session tempo and the other apps are untouched. It needs a live session, so with Link off, alone or without a valid tempo the answer is 409 NO_CLOCK and nothing changes.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Settings after the change (barOffsetBeats is the new offset)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 Link is not connected to another app, so there is no beat to mark (code NO_CLOCK).

503 The Ableton Link service is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 409 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "unavailableReason": "string",
        "enabled": true,
        "quantum": 1,
        "phaseOffsetMs": -500,
        "barOffsetBeats": 31,
        "clockSource": 0,
        "clockSourceName": "audio",
        "beatPolicy": 0,
        "beatPolicyName": "linkGrid",
        "mirrorLinkBeatsToLegacySignal": true,
        "beatWindowMs": 10,
        "tempoSmoothingMs": 2000,
        "fallbackToAudioOnDisconnect": true,
        "beatsDuringSilence": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Beat

The beat clock: the musical time base that effects and cues build on the Ableton Link session (beat line, position in the bar and the phrase, phrase anchor, effect tempo multiplier). Read-mostly: it has no clock of its own, so with Link off or alone in the session it reports valid false.

Get the live state of the beat clock

GET /beat/status

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Beat clock state

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503 The beat clock is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "valid": true,
        "bpm": 0,
        "effectBpm": 0,
        "tempoMultiplier": 0.25,
        "quantum": 1,
        "beatInBar": 0,
        "bar": 0,
        "beatsSinceAnchor": 0,
        "barPhase": 0,
      + "phraseAnchor": { … },
      + "timeline": { … }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the beat clock settings

GET /beat/config

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Beat clock settings

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data >	object

required	Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503** The beat clock is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "tempoMultiplier": 0.25
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change the beat clock settings

PATCH /beat/config

Persists the change and applies it live. The answer is the resulting settings.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `tempoMultiplier` — number
  Enum: `0.25` `0.5` `1` `2` `4`

Responses

200 Settings after the change

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` > (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` (required) — object or null
  Always null on success.
- `meta` > (required) — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422
Not a JSON object, unknown field, wrong type or a value that is not one of the five multipliers (code INVALID_PAYLOAD; the message names the field), or nothing to change (code EMPTY_PATCH).

503 The beat clock is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "tempoMultiplier": 0.25
}
```

Response samples

200 | 400 | 401 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "tempoMultiplier": 0.25
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Mark the start of the phrase

POST /beat/anchor

The downbeat nearest to now becomes bar 0 and bars are counted from there (a little before or after the downbeat is fine). Changing barOffsetBeats or the quantum, or losing the session, puts the anchor back to automatic. Without a beat clock the answer is 409 NO_CLOCK and nothing changes.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Beat clock state after marking the anchor

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409
There is no beat clock (Link is not connected to another app), so there is nothing to anchor (code NO_CLOCK).

503 The beat clock is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 409 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
- "data": {
      "valid": true,
      "bpm": 0,
      "effectBpm": 0,
      "tempoMultiplier": 0.25,
      "quantum": 1,
      "beatInBar": 0,
      "bar": 0,
      "beatsSinceAnchor": 0,
      "barPhase": 0,
    + "phraseAnchor": { … },
    + "timeline": { … }
  },
  "error": { },
- "meta": {
      "apiVersion": "v1",
      "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
      "timestamp": "2019-08-24T14:15:22Z",
      "revision": 0
  }
}
```


Go back to the automatic phrase anchor

DELETE `/beat/anchor`

Bars are counted again from the bar in progress. Harmless when there is no beat clock.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Beat clock state after clearing the anchor

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503** The beat clock is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "valid": true,
        "bpm": 0,
        "effectBpm": 0,
        "tempoMultiplier": 0.25,
        "quantum": 1,
        "beatInBar": 0,
        "bar": 0,
        "beatsSinceAnchor": 0,
        "barPhase": 0,
      + "phraseAnchor": { … },
      + "timeline": { … }
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Timecode

Timecode sync (MTC over MIDI, LTC over an audio channel, or an internal manual clock) and the cue list that fires ActionDispatcher actions when the timecode crosses a cue. Live status, device settings, manual transport and cue CRUD/import/export.

Get the whole cue list

GET `/timecode/cues`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Cue list

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data > required	object Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	

meta > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 | 401

Content type: application/json

Copy | Expand all | Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change the master switch and/or replace the whole cue list

PATCH /timecode/cues

Applies live and persists (the same store as the desktop Timecode Cues editor). Cue items are validated before anything is applied.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `enabled` — boolean
- `entries` > — Array of objects (TimecodeCueInput)

Responses

200 The cue list after the change

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` > required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` > required — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422
Empty patch (code EMPTY_PATCH), entries not an array or a cue that fails validation (code INVALID_PAYLOAD; the message says which field).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "enabled": true,
  - "entries": [
      + { … }
    ]
}
```


Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the cues

GET /timecode/cues/entries

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Cues, in firing-list order

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "entries": [ … ]
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Create a cue

POST `/timecode/cues/entries`

The id is always generated by the server (one sent in the body is dropped). enabled defaults to true and frames to 0.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- **enabled** — boolean
  Default: `true`
- **name** — string
  Optional label shown in the desktop editor.
- **hours** (required) — integer `[ 0 .. 23 ]`
- **minutes** (required) — integer `[ 0 .. 59 ]`
- **seconds** (required) — integer `[ 0 .. 59 ]`
- **frames** — integer `[ 0 .. 29 ]`
  Default: `0`
- **action** (required) — string
  Enum: `"spawnShape"` `"style"` `"artist"` `"imagePlaylist"` `"avatarPlaylist"` `"videoPlaylist"` `"playlistStep"` `"dmxScene"` `"layerVisibility"` `"maskStep"`

	`"setConfigValue"` `"setFromInput"` `"beatAnchor"` `"markDownbeat"`
target	string What the action applies to (style name, artist, playlist, scene, layer...); depends on action.
value	any Action argument; type depends on action. beatAnchor: mark (default) or clear. markDownbeat takes neither a target nor a value.
direction	any playlistStep only: which way to step.
property name* additional property	any

Responses

> **201** The created cue, with its id

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "enabled": true,
    "name": "string",
    "hours": 23,
    "minutes": 59,
```

```
    "seconds": 59,
    "frames": 0,
    "action": "spawnShape",
    "target": "string",
    "value": null,
    "direction": null
}
```


Response samples

201 | 400 | 401 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
        "name": "string",
        "hours": 23,
        "minutes": 59,
        "seconds": 59,
        "frames": 0,
        "action": "spawnShape",
        "target": "string",
        "value": null,
        "direction": null,
        "id": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Reorder the cues

PATCH `/timecode/cues/entries/order`

Registered before the {entryId} route so this exact path is matched instead of being captured as an id.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `order` required — Array of strings
  Every current cue id, each exactly once, in the new order.

Responses

200 The reordered cues

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` required — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 'order' isn't a valid permutation of current ids (code INVALID_ORDER).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "order": [
        "string"
    ]
}
```


Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
     + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change some fields of a cue

PATCH `/timecode/cues/entries/{entryId}`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

`entryId` required	string

REQUEST BODY SCHEMA: application/json required

`enabled`	boolean
`name`	string
`hours`	integer [0 .. 23]
`minutes`	integer [0 .. 59]
`seconds`	integer [0 .. 59]
`frames`	integer [0 .. 29]
`action`	string Enum: `"spawnShape"` `"style"` `"artist"` `"imagePlaylist"` `"avatarPlaylist"` `"videoPlaylist"` `"playlistStep"` `"dmxScene"` `"layerVisibility"` `"maskStep"` `"setConfigValue"` `"setFromInput"` `"beatAnchor"` `"markDownbeat"`
`target`	string
`value`	any
`direction`	any
`property name*` additional property	any

Responses

200 The cue after the change

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object (TimecodeCue) Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422**
Nothing to change once id is ignored (code EMPTY_PATCH), or the merged cue fails validation (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy


```
{
    "enabled": true,
    "name": "string",
    "hours": 23,
    "minutes": 59,
    "seconds": 59,
    "frames": 29,
    "action": "spawnShape",
    "target": "string",
```

```
    "value": null,
    "direction": null
}
```


Response samples

200 | 400 | 401 | 404 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
        "name": "string",
        "hours": 23,
        "minutes": 59,
        "seconds": 59,
        "frames": 0,
        "action": "spawnShape",
        "target": "string",
        "value": null,
        "direction": null,
        "id": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Delete a cue

DELETE /timecode/cues/entries/{entryId}

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

entryId required	string

Responses

200 Deleted

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "deleted": true,
        "id": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Fire a cue's action now

POST `/timecode/cues/entries/{entryId}/execute`

Runs the same synchronous ActionDispatcher path as POST /actions/execute, ignoring the incoming timecode; handy to test a new cue. The dispatch context carries source "timecode" and the cue id.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

entryId required	string

Responses

200 Action dispatched

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`

data >	required	object Unconstrained here; each operation's response narrows this via allOf.
error	required	object or null Always null on success.
meta >	required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

422
The cue's action was refused by the dispatcher (code ACTION_FAILED; the message is the dispatcher's reason).

503
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 404 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "dispatched": true,
        "entryId": "string",
        "action": "string",
        "target": "string"
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the saved cue files

GET /timecode/cues/files

For a client with no filesystem access of its own (a phone) that wants a picker; rebuild the path as timecode_cues/.json for the file routes below.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 File base names (empty when the folder does not exist)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "files": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Save the cue list to a JSON file

POST /timecode/cues/file/export

Writes {enabled, entries} to disk, the server-side counterpart of the desktop editor's Save. Accepts an arbitrary path (relative to the application folder, or absolute), like the editor.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

path string

Responses

200 What was written, with the resolved path

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object (TimecodeCuesFileResult) Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

500 Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy

```
{
    "path": "string"
}
```

Response samples

200 400 401 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ],
        "path": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Load the cue list from a JSON file

POST `/timecode/cues/file/import`

Replaces the whole live cue list with the file's {enabled, entries}, the server-side counterpart of the desktop editor's Load. Every entry is validated first; nothing is applied if one fails.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

`path` string

Responses

200 The cue list now in force, with the resolved path

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object (TimecodeCuesFileResult) Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422**
The file is not a JSON object with an entries array (code INVALID_JSON, although the request body was fine) or an entry fails validation (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "path": "string"
}
```

Response samples

200 400 401 404 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ],
        "path": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the live decoded timecode

GET /timecode/status

Read-only; what is being decoded right now, as opposed to /timecode/config (what can be configured).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Timecode status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
- "data": {
      "enabled": true,
      "sourceMode": 0,
      "activeSource": "",
      "connected": true,
      "timecode": "00:12:34:05",
      "hours": 0,
      "minutes": 0,
      "seconds": 0,
      "frames": 0,
      "dropFrame": true,
      "reversed": true,
      "fps": 0,
      "frameRateLabel": "29.97 fps (DF)",
      "totalSeconds": 0,
      "ltcSignalConfidence": 0
  },
  "error": { },
- "meta": {
      "apiVersion": "v1",
      "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
      "timestamp": "2019-08-24T14:15:22Z",
      "revision": 0
  }
}
```


Get the Timecode settings and the available devices

GET /timecode/config

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Timecode settings

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
        "sourceMode": 0,
        "selectedMidiInput": -1,
      + "availableMidiInputs": [ … ],
        "selectedAudioInput": -1,
      + "availableAudioInputs": [ … ],
        "ltcChannelIndex": 0
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Change some of the Timecode settings

PATCH `/timecode/config`

Same store as the desktop Settings > Timecode panel; applies live. See TimecodeConfigPatch for the order and the fact that it is not atomic.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `sourceMode` integer [0 .. 4]
- `selectedMidiInput` integer >= -1
  Index in availableMidiInputs, -1 = none.
- `selectedAudioInput` integer >= -1
  Index in availableAudioInputs, -1 = none.
- `ltcChannelIndex` integer >= 0
- `enabled` boolean

Responses

200 Settings after the change

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
The active edition does not include this feature (code EDITION_REQUIRED; the message names the edition). See x-edition on the operation.

> **422**
Empty patch (code EMPTY_PATCH), or a field of the wrong type or out of range (code INVALID_PAYLOAD; the message names it).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "sourceMode": 4,
    "selectedMidiInput": -1,
    "selectedAudioInput": -1,
    "ltcChannelIndex": 0,
    "enabled": true
```

```
}
```


Response samples

200 | 400 | 401 | 403 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
        "sourceMode": 0,
        "selectedMidiInput": 0,
        "selectedAudioInput": 0,
        "ltcChannelIndex": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Rescan the MIDI input ports

POST /timecode/config/refresh-midi-inputs

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 The refreshed port list

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 401 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
     + "availableMidiInputs": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Rescan the audio input devices

POST `/timecode/config/refresh-audio-inputs`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 The refreshed device list

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "availableAudioInputs": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the manual clock transport

GET /timecode/manual

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 Manual transport

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

503

The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "running": true,
        "frameRate": 0
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Play / pause the manual clock or change its frame rate

PATCH `/timecode/manual`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `running` — boolean
- `frameRate` — number > 0

Responses

200 Transport after the change

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` required — object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422**
Empty patch (code EMPTY_PATCH), running not a boolean or frameRate not a positive number (code INVALID_PAYLOAD).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "running": true,
    "frameRate": 0
}
```

Response samples

200 | 400 | 401 | 422 | 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "running": true,
        "frameRate": 0
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Jump the manual clock to an absolute position

POST /timecode/manual/locate

Does not fire the cues it skips (the next position is not compared against the previous one).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

hours required	integer	
minutes required	integer	
seconds required	integer	
frames required	integer	

Responses

200 The new position

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	

data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> 400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> 401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> 422

A field is missing or not a number (code INVALID_PAYLOAD), or sourceMode is not 4 (code WRONG_SOURCE_MODE).

> 503

The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "hours": 0,
    "minutes": 0,
    "seconds": 0,
    "frames": 0
}
```

Response samples

200 400 401 422 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "totalSeconds": 0,
        "timecode": "00:12:34:05"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Jump the manual clock relative to its position

POST `/timecode/manual/jump`

Does not fire the cues it skips (the next position is not compared against the previous one).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

`deltaSeconds`
required

number
Relative jump: positive = forward, negative = back (clamped at 00:00:00:00).

Responses

200 The new position

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422**
deltaSeconds is missing or not a number (code INVALID_PAYLOAD), or sourceMode is not 4 (code WRONG_SOURCE_MODE).

> **503**
The timecode provider (or, for a cue execute, the action dispatcher) is not wired into the server (code NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy

```
{
    "deltaSeconds": 0
}
```

Response samples

200 400 401 422 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "totalSeconds": 0,
        "timecode": "00:12:34:05"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Schedule

Programming schedule: actions fired at a time of day on chosen weekdays (the desktop Schedule editor's data), entry CRUD, JSON file import/export and a simulated clock to test it.

Get the whole programming schedule

GET `/schedule`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 | 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Change the master switch and/or replace the whole entry list

PATCH `/schedule`

Applies live and persists (prefs.programmingSchedule, the same data the desktop Schedule editor edits). entries REPLACES the list; every entry is validated first and one bad entry refuses the patch. An empty object is 422 EMPTY_PATCH.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `enabled` — boolean
- `entries` — Array of objects (ScheduleEntryInput)

Responses

200 OK

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` (required) — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` (required) — object or null
  Always null on success.
- `meta` (required) — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "enabled": true,
  - "entries": [
      + { … }
    ]
}
```


Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
```

```
}
```

List the schedule entries

GET `/schedule/entries`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

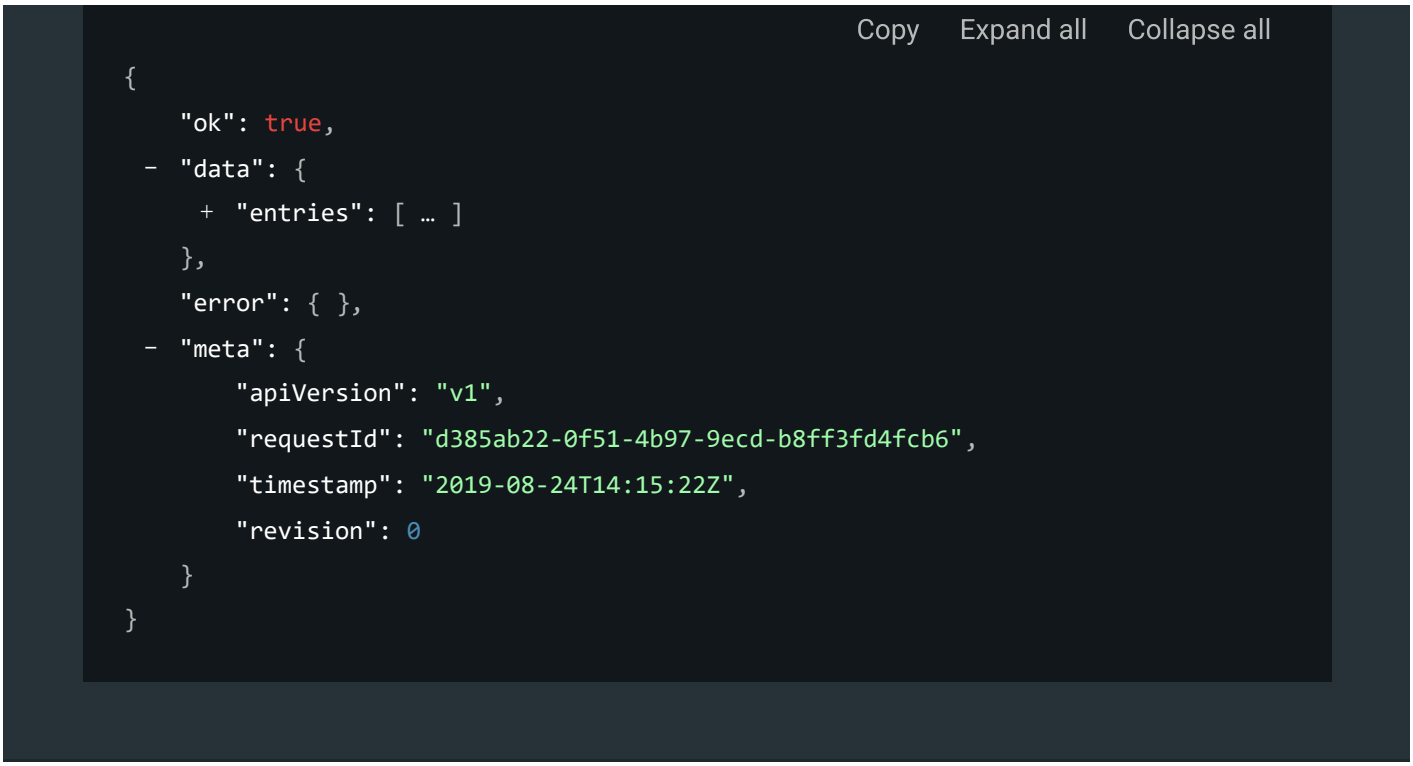
`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 | 401

Content type
application/json



Create a schedule entry

POST `/schedule/entries`

The id is always generated by the server (one sent in the body is dropped). enabled defaults to true and second to 0.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- **enabled** — boolean
  Default: `true`
- **name** — string
  Optional label shown in the desktop editor.
- **hour** *required* — integer `[ 0 .. 23 ]`
- **minute** *required* — integer `[ 0 .. 59 ]`
- **second** — integer `[ 0 .. 59 ]`
  Default: `0`

daysOfWeek required	Array of integers non-empty [items [0 .. 6]] Days it fires, Sunday = 0.
action required	string Enum: `"spawnShape"` `"style"` `"artist"` `"imagePlaylist"` `"avatarPlaylist"` `"videoPlaylist"` `"playlistStep"` `"dmxScene"` `"layerVisibility"` `"maskStep"` `"setConfigValue"` `"setFromInput"` `"beatAnchor"` `"markDownbeat"`
target	string
value	any Action argument; type depends on action.
direction	any playlistStep only: which way to step.
property name* additional property	any

Responses

> **201** The created entry, with its id

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "enabled": true,
    "name": "Opening style",
    "hour": 21,
    "minute": 30,
    "second": 0,
  - "daysOfWeek": [
        5,
        6
    ],
    "action": "style",
    "target": "Neon Tunnel"
}
```


Response samples

201 | 400 | 401 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "id": "5b0c8a52-6f0e-4e43-9a3b-0c1d2e3f4a5b",
        "enabled": true,
        "name": "Opening style",
        "hour": 21,
        "minute": 30,
        "second": 0,
      + "daysOfWeek": [ … ],
        "action": "style",
        "target": "Neon Tunnel"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Reorder the schedule entries

PATCH `/schedule/entries/order`

Registered before the {entryId} route so this exact path is matched instead of being captured as an id.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `order` required — Array of strings
  Every current entry id, each exactly once, in the new order.

Responses

200 The reordered entries

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` required — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** 'order' isn't a valid permutation of current ids (code INVALID_ORDER).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "order": [
        "string"
    ]
}
```


Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "entries": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Change some fields of a schedule entry

PATCH `/schedule/entries/{entryId}`

Any subset of the entry fields, merged into the stored entry (id is ignored). The merged entry is validated. Nothing to change once id is ignored is 422 EMPTY_PATCH.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

`entryId` required	string

REQUEST BODY SCHEMA: application/json required

`enabled`	boolean
`name`	string
`hour`	integer [0 .. 23]
`minute`	integer [0 .. 59]
`second`	integer [0 .. 59]
`daysOfWeek`	Array of integers non-empty [items [0 .. 6]]
`action`	string
`target`	string
`value`	any
`property name*` additional property	any

Responses

200 The entry after the change

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object (ScheduleEntry) Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **404** Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "enabled": true,
    "name": "string",
    "hour": 23,
    "minute": 59,
    "second": 59,
  - "daysOfWeek": [
        6
    ],
    "action": "string",
    "target": "string",
```

```
    "value": null
}
```


Response samples

200 | 400 | 401 | 404 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "id": "5b0c8a52-6f0e-4e43-9a3b-0c1d2e3f4a5b",
        "enabled": true,
        "name": "Opening style",
        "hour": 21,
        "minute": 30,
        "second": 0,
      + "daysOfWeek": [ … ],
        "action": "style",
        "target": "Neon Tunnel"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Delete a schedule entry

DELETE `/schedule/entries/{entryId}`

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

entryId required	string

Responses

200 Deleted

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data › required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta › required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

Response samples

200 | 401 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "deleted": true,
        "id": "string"
    },
```

```
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Fire a schedule entry's action now

POST `/schedule/entries/{entryId}/execute`

Runs the entry's action through the same synchronous ActionDispatcher path as POST /actions/execute, ignoring its hour/minute/days. The dispatch context carries source "schedule" and the entry id.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

PATH PARAMETERS

- `entryId` — required — string

Responses

200 Action dispatched

RESPONSE SCHEMA: application/json

- `ok` — required — boolean
  Value: `true`
- `data` > — required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` — required — object or null
  Always null on success.

meta > required — object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

422
The action was refused by the dispatcher (code ACTION_FAILED; the message is the dispatcher's reason).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 404 | 422 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "dispatched": true,
        "entryId": "5b0c8a52-6f0e-4e43-9a3b-0c1d2e3f4a5b",
        "action": "style",
        "target": "Neon Tunnel"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Save the schedule to a JSON file

POST `/schedule/file/export`

Writes {enabled, entries} to disk, the server-side counterpart of the desktop editor's Export. path defaults to programming_schedule/programming_schedule.json; a relative path is resolved against the application folder and an absolute one is used as is.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

- `path` — string

Responses

200 What was written, with the resolved path

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` > (required) — object (ScheduleFileResult)
  Unconstrained here; each operation's response narrows this via allOf.
- `error` (required) — object or null
  Always null on success.
- `meta` > (required) — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy


```
{
    "path": "string"
}
```


Response samples

200 | 400 | 401 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ],
        "path": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Load the schedule from a JSON file

POST `/schedule/file/import`

Replaces the whole live schedule with the file's {enabled, entries}. Every entry is validated first; nothing is applied if one fails. A file that is not a JSON object with an entries array is 422 INVALID_JSON.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
optional

- `path` — string

Responses

200 The schedule now in force, with the resolved path

RESPONSE SCHEMA: application/json

- `ok` (required) — boolean
  Value: `true`
- `data` > (required) — object (ScheduleFileResult)
  Unconstrained here; each operation's response narrows this via allOf.
- `error` (required) — object or null
  Always null on success.
- `meta` > (required) — object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy


```
{
    "path": "string"
}
```


Response samples

200 | 400 | 401 | 404 | 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "enabled": true,
      + "entries": [ … ],
        "path": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Get the simulated-clock offset of the scheduler

GET `/schedule/clock`

Runtime only: the offset is never persisted, so a forgotten offset cannot survive a restart and fire the real schedule at the wrong time.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "offsetSeconds": 3600,
        "realNow": "2026-10-13T20:30:00",
        "simulatedNow": "2026-10-13T21:30:00"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Set or shift the simulated-clock offset

PATCH `/schedule/clock`

Send offsetSeconds (absolute) and/or jumpSeconds (relative, added after offsetSeconds when both are sent). One of the two is required. Useful to test a schedule without waiting for its hour.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `offsetSeconds` — number
  Absolute offset.
- `jumpSeconds` — number
  Relative shift of the current offset.

Responses

200 The clock after the change

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "jumpSeconds": 600
}
```

Response samples

200 | 400 | 401 | 422

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "offsetSeconds": 3600,
        "realNow": "2026-10-13T20:30:00",
        "simulatedNow": "2026-10-13T21:30:00"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Actions

Run one ActionDispatcher action now, the same path the schedule, the timecode cues and the MIDI/OSC/HID controllers use.

Run an ActionDispatcher action now

POST `/actions/execute`

Fire-and-forget: the same synchronous path the Programming Schedule, the timecode cues and the MIDI/OSC/HID controllers use, so an action launched here behaves identically. The dispatch context carries source "api". The vocabulary of action/target/value is the one of the Programming Schedule entries.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

`action` string
required

Enum: `"spawnShape"` `"style"` `"artist"` `"imagePlaylist"` `"avatarPlaylist"` `"videoPlaylist"` `"playlistStep"` `"dmxScene"` `"layerVisibility"` `"maskStep"` `"setConfigValue"` `"setFromInput"` `"beatAnchor"` `"markDownbeat"`

target
string
What the action applies to (style name, artist, playlist, scene, layer...); depends on action.

value
any
Action argument; type depends on action.

params >
object
Extra named arguments (playlistStep reads direction here when value is empty).

Responses

200 OK

RESPONSE SCHEMA: application/json

ok
required
boolean
Value: `true`

data >
required
object
Unconstrained here; each operation's response narrows this via allOf.

error
required
object or null
Always null on success.

meta >
required
object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422
The action was refused by the dispatcher (code ACTION_FAILED; the message is the dispatcher's reason).

> 503

The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Request samples

Payload

Content type
application/json

Copy

```
{
    "action": "style",
    "target": "Neon Tunnel"
}
```

Response samples

200 | 400 | 401 | 422 | 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "dispatched": true,
        "action": "style",
        "target": "Neon Tunnel"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


NDI

NDI output through the separate DjMirrorPulseNdiBridge process: the layout file it follows (outputs[]) and the control of the process itself (start/stop/status/settings).

Read an NDI layout file

GET /ndi/config

Returns the file's own JSON object plus path. Unlike the Spout layout GET it does not parse outputs[]: the bridge process is the tolerant reader. A file that exists but is not valid JSON is 500 INVALID_JSON.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

path — string
Layout file, relative to the application folder or absolute. Default ndi_config/ndi_bridge_config.example.json.

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

500 Local read/write failure (code INTERNAL_ERROR).

Response samples

200 | 401 | 404 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "path": "string",
      + "outputs": [ … ]
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Write an NDI layout file

PUT /ndi/config

outputs must be present and an array, with no per-field validation. Written exactly as sent. DjMirror Pulse holds no live state for it: the bridge re-reads the file by itself while it runs.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

QUERY PARAMETERS

path — string
Layout file, relative to the application folder or absolute. Default ndi_config/ndi_bridge_config.example.json.

REQUEST BODY SCHEMA: application/json
required

outputs > required — Array of objects (LayoutOutput)

property name* additional property — any

Responses

200 OK

RESPONSE SCHEMA: application/json

ok	required	boolean Value: `true`
data >	required	object Unconstrained here; each operation's response narrows this via allOf.
error	required	object or null Always null on success.
meta >	required	object (Meta)

> **400** Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **422** Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

> **500** Local read/write failure (code INTERNAL_ERROR).

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "outputs": [
      + { … }
    ]
}
```


Response samples

200 400 401 422 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "path": "string",
      + "outputs": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the state of the NDI bridge process

GET /ndi/status

Process-only: the bridge has no IPC channel back to the app, so this means "the process is alive", never frames sent or receivers connected.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error	object or null

required	Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 | 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "running": true,
        "pid": 18420,
        "startedAt": "2026-10-13T10:15:00Z",
        "lastError": null,
        "autoStart": false,
        "statusDetail": "process-only"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Start the NDI bridge process

POST /ndi/start

No body: it always uses what GET /ndi/settings has. Broadcasts ndi.started.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Started; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403
The active edition does not include this feature (code EDITION_REQUIRED; the message names the edition). See x-edition on the operation.

409
Already running (code ALREADY_RUNNING). The running process is not touched. A bridge started by hand from a terminal is not detected.

500 The bridge process could not be launched (code START_FAILED; the message says why).

503
The service this route needs is not wired into the server or not available on this build (code NOT_AVAILABLE; the message says which).

Response samples

200 | 401 | 403 | 409 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "available": true,
        "running": true,
        "pid": 18420,
        "startedAt": "2026-10-13T10:15:00Z",
        "lastError": null,
        "autoStart": false,
        "statusDetail": "process-only"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Stop the NDI bridge process

POST /ndi/stop

Abrupt: a kill(), not an orderly shutdown (the windowless process ignores terminate()). Broadcasts ndi.stopped.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 Stopped; the body is the status

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

409 Not running (code NOT_RUNNING).

Response samples

200 | 401 | 409

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
```

```
  - "data": {
        "available": true,
        "running": true,
        "pid": 18420,
        "startedAt": "2026-10-13T10:15:00Z",
        "lastError": null,
        "autoStart": false,
        "statusDetail": "process-only"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Get the NDI bridge launch settings

GET /ndi/settings

Not the layout file's content: that is /ndi/config.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.

error	required	object or null Always null on success.
meta >	required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "autoStart": false,
        "bridgeExecutablePath": "",
        "effectiveExecutablePath": "C:/DjMirrorPulse/DjMirrorPulseNdiBridge.exe",
        "singleWindowMode": true,
        "windowTitle": "",
        "sourceName": "DjMirror Pulse Main",
        "fps": 0,
        "scale": 1,
        "width": 0,
        "height": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Change the NDI bridge launch settings

PATCH `/ndi/settings`

Persists across restarts and never starts or stops the bridge by itself: call POST /ndi/start after. An empty patch is 422 EMPTY_PATCH. Broadcasts ndi.updated (at most once per 150 ms).

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

REQUEST BODY SCHEMA: application/json
required

- `autoStart` — boolean
  Launch the bridge when the application starts.
- `bridgeExecutablePath` — string
  For a nonstandard install. Empty = auto-detect.
- `singleWindowMode` — boolean
  false = the next start follows the layout file; true = one uncropped source for the whole window.
- `windowTitle` — string
  singleWindowMode only. Empty = the bridge's default.
- `sourceName` — string
  singleWindowMode only. Empty = the bridge's default.
- `fps` — integer
  singleWindowMode only. 0 = the bridge's default (60).
- `scale` — number [0.1 .. 1]
  singleWindowMode only.
- `width` — integer [16 .. 16384]
  singleWindowMode only; set together with height. 0 = follow the window.
- `height` — integer [16 .. 16384]
  singleWindowMode only; set together with width.

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

400 Body is not parseable JSON (code INVALID_JSON, hint carries the parser error).

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

422 Body parses but has the wrong shape/required field (code INVALID_PAYLOAD).

Request samples

Payload

Content type
application/json

Copy

```
{
    "singleWindowMode": true,
    "sourceName": "Main",
    "scale": 0.5
}
```

Response samples

200 400 401 422

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "autoStart": false,
        "bridgeExecutablePath": "",
        "effectiveExecutablePath": "C:/DjMirrorPulse/DjMirrorPulseNdiBridge.exe",
        "singleWindowMode": true,
        "windowTitle": "",
        "sourceName": "DjMirror Pulse Main",
        "fps": 0,
        "scale": 1,
        "width": 0,
        "height": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the saved NDI layout files

GET /ndi/files

Base names (no extension) of the .json files under ndi_config/, to build a ?path= picker. Empty, not an error, before the folder has anything in it.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth*

Responses

200 OK

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "files": [ … ]
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Realtime

Long-poll config change subscription.

Long-poll for the next config change

GET `/config/subscribe`

Held open until the next config change; the server then writes a single event and the connection closes (client must re-subscribe). NOT wrapped in the standard ok/data envelope on success -- see SubscribeEvent. DMX-related changes are not delivered here; connect the WebSocket (ws://localhost:<httpPort+1>) for those.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth*

Responses

200 A single config.changed event, then the connection closes

RESPONSE SCHEMA: application/json

`type` required	string	Value: `"event"`
`eventType` required	string	Value: `"config.changed"`
`resource` required	string	Value: `"config"`
`action` required	string	Value: `"updated"`
`data` > required	object	

requestId required	string
timestamp required	string <date-time>

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "type": "event",
    "eventType": "config.changed",
    "resource": "config",
    "action": "updated",
    "data": { },
    "requestId": "string",
    "timestamp": "2019-08-24T14:15:22Z"
}
```


Pairing

Mobile app pairing by QR code (docs_tech/docs_mobile/PLAN_MOBILE_QR_PAIRING.md). The phone has no license of its own: it claims a one-use code shown on the desktop, gets a deviceToken and inherits the desktop license. Management routes (QR, confirmation, device list) need the API key when there is one; without an API key they are only accepted from the computer itself (loopback). A deviceToken never manages pairing (403 FORBIDDEN).

Pair a phone with the code of the QR

POST `/pairing/claim`

Public: the one-use code authenticates it. The code expires after 3 minutes and is locked after 5 wrong attempts; at most 10 claims per minute per address. If the desktop asks for confirmation (the default), the response waits up to 30 s for the user to accept.

REQUEST BODY SCHEMA: application/json
required

- `code` — string
  26-character base32 Crockford code from the QR (c=).
- `shortCode` — string
  XXXX-XXXX code shown under the QR.
- `deviceId` required — string <= 64 characters `^[A-Za-z0-9_.{}-]+$`
  Stable id of the phone install.
- `deviceName` — string
- `platform` — string
- `appVersion` — string
- `protocol` required — integer
  Value: `1`

Responses

200 Paired

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` > required — object (PairingClaimData)
  Unconstrained here; each operation's response narrows this via allOf.

error
required

object or null

Always null on success.

meta >
required

object (Meta)

> 400

Not a JSON object (INVALID_JSON), no code nor shortCode (INVALID_PAYLOAD), protocol other than 1 (UNSUPPORTED_PROTOCOL, details.supported) or a bad deviceId (INVALID_DEVICE).

> 403

No active license or trial (LICENSE_INACTIVE), wrong code (PAIRING_CODE_INVALID, counts towards the lock), the edition admits no more phones (MOBILE_LIMIT_REACHED, details.limit/used/edition; the code is not spent) or rejected on the desktop (PAIRING_REJECTED).

> 404

No code on the desktop: never created, cancelled or already used (PAIRING_CODE_UNKNOWN).

> 408

Nobody confirmed on the desktop within 30 s (CONFIRMATION_TIMEOUT); the code is cancelled.

> 409 Another phone is waiting for confirmation (PAIRING_PENDING).

> 410 The code expired, or a new one was generated while waiting (PAIRING_CODE_EXPIRED).

> 423 5 wrong codes: generate a new one (PAIRING_LOCKED).

> 429 More than 10 claims in a minute from that address (RATE_LIMITED).

> 503 Pairing not available (NOT_AVAILABLE) or the desktop is shutting down (SHUTTING_DOWN).

Request samples

Payload

Content type
application/json

Copy

```
{
    "code": "string",
    "shortCode": "7K3M-Q9TZ",
```

```
    "deviceId": "^A$",
    "deviceName": "Pixel 8",
    "platform": "android",
    "appVersion": "string",
    "protocol": 1
}
```


Response samples

200 | 400 | 403 | 404 | 408 | 409 | 410 | 423 | 429 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "device": { … },
      + "desktop": { … },
      + "license": { … },
      + "attestation": { … },
        "deviceToken": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Identify the desktop

GET /pairing/info

Public. Lets a paired phone recognise its desktop after an IP change, without revealing anything the QR does not already say.

Responses

200 Desktop identity

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

503 Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "protocol": 1,
        "id": "string",
        "name": "string"
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


The paired phone's link and the desktop license

GET /mobile/link

Always needs the phone's deviceToken, also without an API key. Records when and from where the phone was last seen.

AUTHORIZATIONS: > *deviceToken*

Responses

200 Link

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data > required	object Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

401

Missing or unknown deviceToken (code UNAUTHORIZED), or the phone was unpaired on the computer (code DEVICE_REVOKED).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "device": { … },
      + "desktop": { … },
      + "license": { … },
      + "attestation": { … }
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


The phone unpairs itself

DELETE /mobile/link

AUTHORIZATIONS: > *deviceToken*

Responses

200 Unpaired

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

401
Missing or unknown deviceToken (code UNAUTHORIZED), or the phone was unpaired on the computer (code DEVICE_REVOKED).

503 Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "unpaired": true
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Create a pairing QR code

POST `/pairing/sessions`

Replaces the previous code, if any. The body is optional.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth* or None

REQUEST BODY SCHEMA: application/json
optional

`requireConfirmation` boolean
Default: the desktop setting (Settings > Mobile app).

Responses

> **201** The new code

> **400**
Not a JSON object (INVALID_JSON) or requireConfirmation is not a boolean (INVALID_PAYLOAD).

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
No active license or trial (LICENSE_INACTIVE), or not allowed to manage pairing (FORBIDDEN).

> 503

The HTTP server is not running (SERVER_NOT_RUNNING), no usable network address (NO_NETWORK) or pairing not available (NOT_AVAILABLE).

Request samples

Payload

Content type
application/json

Copy


```
{
    "requireConfirmation": true
}
```


Response samples

201 | 400 | 401 | 403 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "active": true,
        "uri": "djmirror://pair?v=1&h=192.168.1.20&p=8080&c=...&id=1a2b3c4d&n=Stud
        "shortCode": "string",
        "expiresAt": 0,
      + "hosts": [ … ],
        "port": 0,
        "desktopId": "string",
        "requireConfirmation": true
    },
    "error": { },
```

```
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


The current pairing code

GET /pairing/sessions/current

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth* or None

Responses

200 The current code (active false if none)

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`	
data > required	object Unconstrained here; each operation's response narrows this via allOf.	
error required	object or null Always null on success.	
meta > required	object (Meta)	

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "active": true,
        "uri": "djmirror://pair?v=1&h=192.168.1.20&p=8080&c=...&id=1a2b3c4d&n=Stud
        "shortCode": "string",
        "expiresAt": 0,
      + "hosts": [ … ],
        "port": 0,
        "desktopId": "string",
        "requireConfirmation": true
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Cancel the current pairing code

DELETE /pairing/sessions/current

A phone waiting for confirmation gets 410 PAIRING_CODE_EXPIRED.

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth* or None

Responses

200 Cancelled

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

503 Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 503

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true,
  - "data": {
        "active": false
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


A phone waiting for confirmation

GET /pairing/pending

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth* or None

Responses

200 Pending claim

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.
error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "pending": true,
        "deviceName": "string",
        "platform": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


Approve or reject the waiting phone

POST /pairing/pending/{action}

AUTHORIZATIONS: > *bearerAuth* or *apiKeyAuth* or None

PATH PARAMETERS

- `action` required — string
  Enum: `"approve"` `"reject"`

Responses

200 Done

RESPONSE SCHEMA: application/json

- `ok` required — boolean
  Value: `true`
- `data` > required — object
  Unconstrained here; each operation's response narrows this via allOf.
- `error` required — object or null
  Always null on success.
- `meta` > required — object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

> **404** No phone is waiting (NO_PENDING), or an action other than approve/reject (NOT_FOUND).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 404 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "action": "approve"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```


List the paired phones

GET /pairing/devices

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth* or None

Responses

200 Paired phones and the edition limit

RESPONSE SCHEMA: application/json

ok required	boolean Value: `true`
data > required	object Unconstrained here; each operation's response narrows this via allOf.

error required	object or null Always null on success.
meta > required	object (Meta)

> **401** Missing/invalid bearer token or API key (code UNAUTHORIZED).

> **403**
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
      + "devices": [ … ],
        "limit": 0,
        "used": 0
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```

Unpair a phone

DELETE `/pairing/devices/{deviceId}`

Its token stops working (401 DEVICE_REVOKED) and its WebSocket is closed.

AUTHORIZATIONS: *bearerAuth* or *apiKeyAuth* or None

PATH PARAMETERS

`deviceId` required	string

Responses

200 Unpaired

RESPONSE SCHEMA: application/json

`ok` required	boolean Value: `true`
`data` > required	object Unconstrained here; each operation's response narrows this via allOf.
`error` required	object or null Always null on success.
`meta` > required	object (Meta)

401 Missing/invalid bearer token or API key (code UNAUTHORIZED).

403
Not the computer itself and no API key (or a deviceToken, which never manages pairing) (code FORBIDDEN).

404 Resource/section/id/scene/file doesn't exist (code NOT_FOUND).

> **503** Mobile pairing is not available in this process (code NOT_AVAILABLE).

Response samples

200 | 401 | 403 | 404 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "ok": true,
  - "data": {
        "unpaired": "string"
    },
    "error": { },
  - "meta": {
        "apiVersion": "v1",
        "requestId": "d385ab22-0f51-4b97-9ecd-b8ff3fd4fcb6",
        "timestamp": "2019-08-24T14:15:22Z",
        "revision": 0
    }
}
```