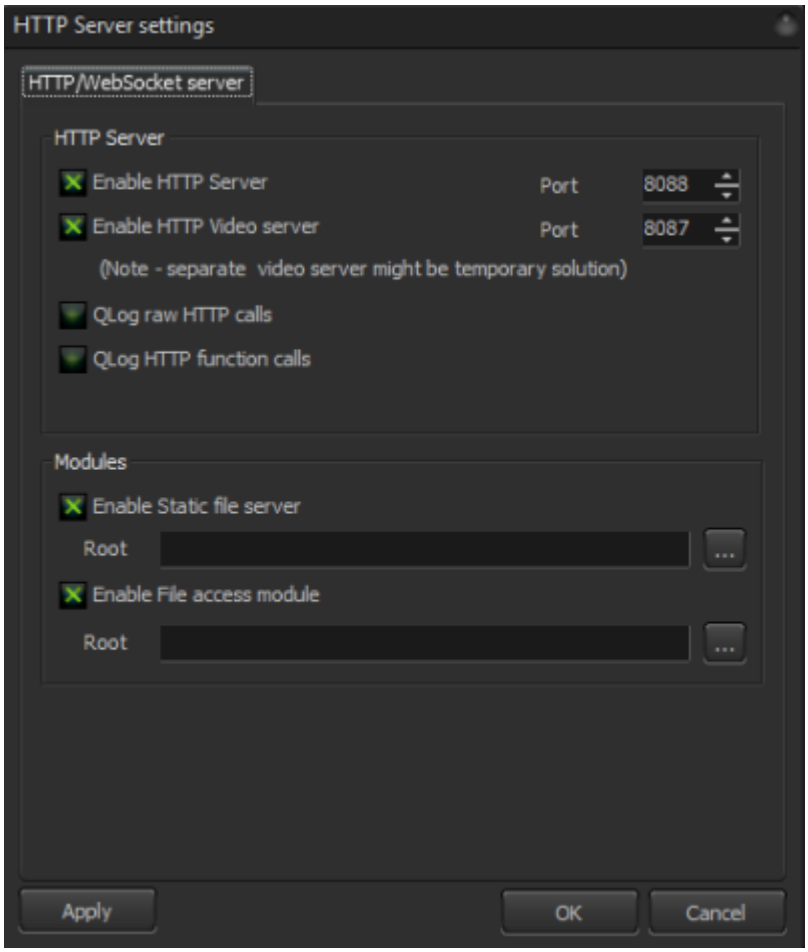
HTTP Server

For the release of BEYOND 5.8, we introduced the HTTP server, available to those with BEYOND Ultimate. The HTTP server is a new way to interact with BEYOND's internal commands and controls, as well as pangoscript and objects. What is possible with this really is at the limits of your creativity, but we imagine people will use this to create their own interfaces for BEYOND and external tools, using more commonly used programming languages and tools compared to pangoscript itself.



Introduction to the HTTP Server

The BEYOND HTTP Server is a REST gateway built into Pangolin BEYOND. It is available only in BEYOND Ultimate. Other editions do not include it and cannot add it as an option. It exposes the live show engine — workspace, cue grid, playlist, live-control surfaces, projection-zone flags, and PangoScript — to any HTTP client on the LAN. There is no cloud service: the API is the BEYOND instance on that machine.

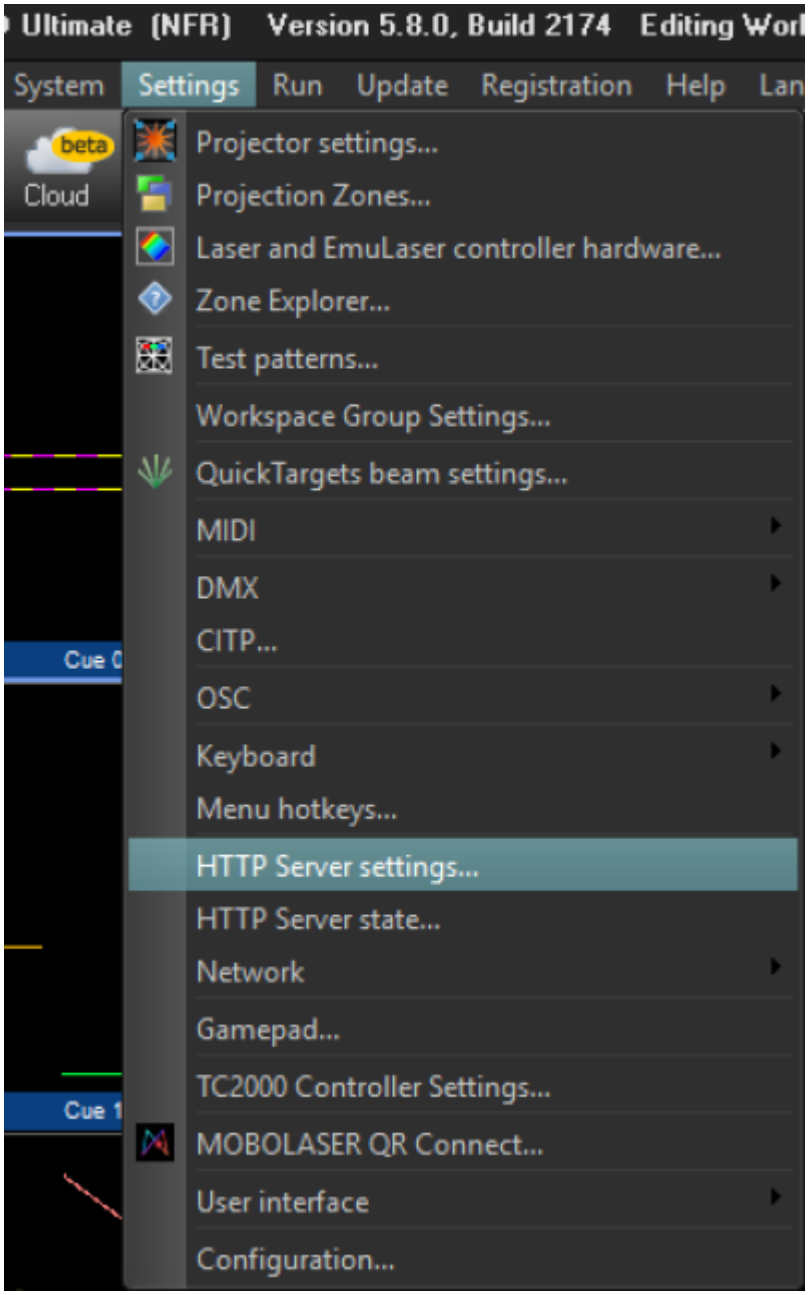
- Base URL: http://<beyond-host>:8088/apiv1/ (use the port configured in Settings).
- Default HTTP port 8088. Video preview is a separate process on 8087.
- Plain HTTP only. Put a reverse proxy in front if you need TLS.

- REST calls do not require credentials in current builds.
- Live Try-it-out is the API Playground on the show machine: http://<beyond-host>:8088/beyondswagger/
- This wiki page is documentation. It does not send live requests.

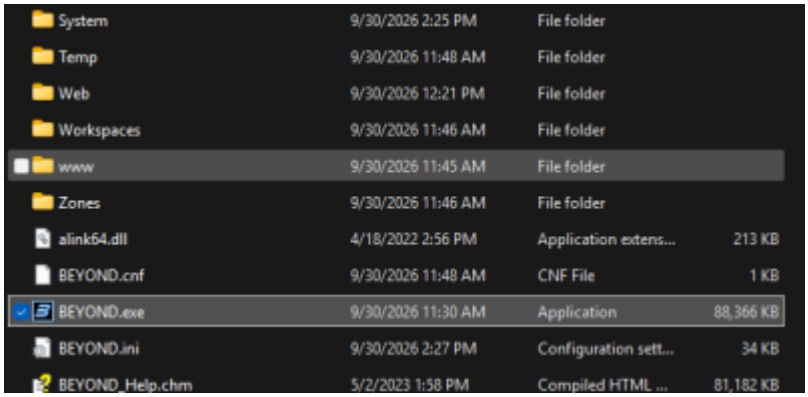
Enable the server and open the examples

The BEYOND Ultimate 5.8 installer places the example pages and API Playground in <BEYOND.exe folder>\www. Leave Root blank to serve that folder.

1. Open BEYOND Ultimate. Settings → HTTP Server settings.
2. Enable HTTP server, set the port (default 8088), click Apply.
3. Leave Static file server → Root blank. Blank Root is <BEYOND.exe folder>\www (stored as HttpStatDir in BEYOND.ini if you later pick a folder).



Layout	What to do	URL
Default (www)	Leave Root blank. Open the pages the installer already placed in <exe>\www.	http://<host>:<port>/index.html
Custom folder	Optional: click … and choose a folder that contains index.html.	http://<host>:<port>/index.html



API Playground: http://<host>:<port>/beyondswagger/

Smoke-test:

```
GET http://127.0.0.1:8088/apiv1/ws/info
```

A 200 response is JSON describing the current workspace.

Request and response format

- REST paths live under /apiv1/. Future breaking changes will use /apiv2/.
- JSON endpoints return application/json unless documented otherwise (thumbnails return images).
- PangoScript to POST /apiv1/ps/execscript is raw UTF-8 text (Content-Type: text/plain). JSON wrappers such as {“code”:“…”} are not supported.
- Treat any non-2xx status as failure.
- execscript does not return WriteLn / echo text.

First script:

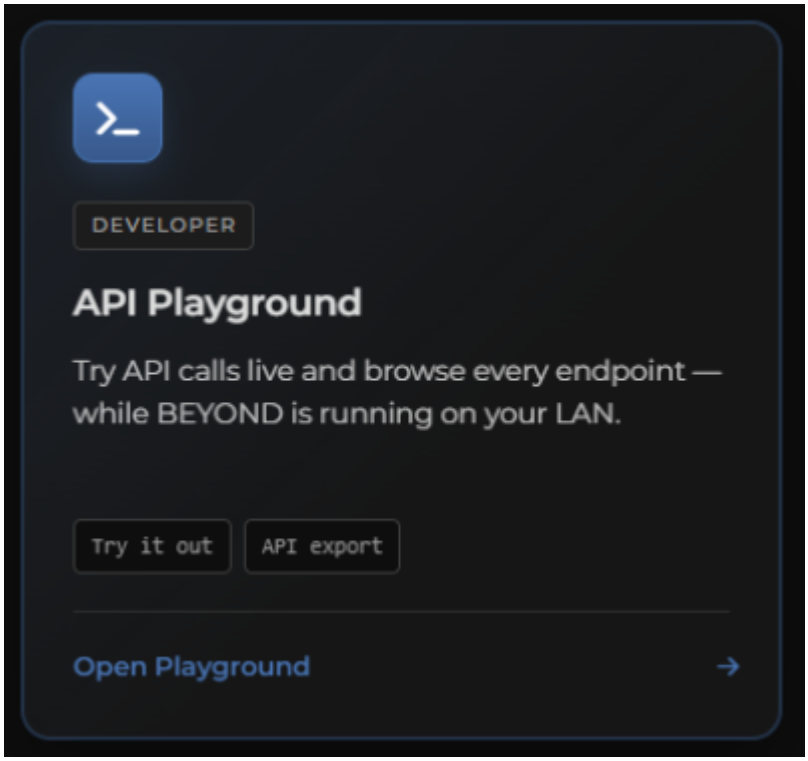
```
POST http://127.0.0.1:8088/apiv1/ps/execscript
Content-Type: text/plain

Blackout
Sleep 500
EnableLaserOutput
```

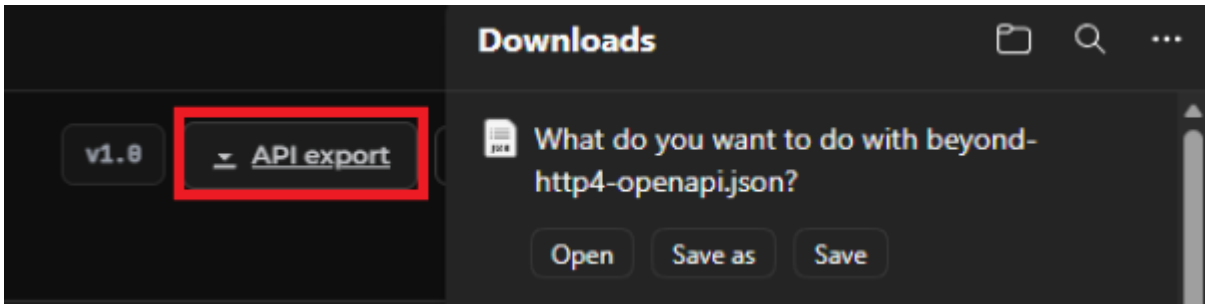
```
curl -X POST http://127.0.0.1:8088/apiv1/ps/execscript \
  -H "Content-Type: text/plain" \
  --data-binary $'Blackout\nSleep 500\nEnableLaserOutput'
```

json File Location

Once the demo file is loaded, head to the API playground from the homepage.



From there at the top right of the API playground, you will see an API Export button, which will cause the json file to download.



Core concepts

PangoScript objects

Most automation flows through PangoScript. Use these names with GET /apiv1/psobject?name=…

Name	Purpose
master	Master Live Control + transition / metro / BPM state
playlist	Playback state mirror
cue.<page>.<index>	One cue by page and index
zone	Zone list (names, preview, uGC, flags). projectorIndex is 1-based.
zone.<n>	Live Control subset for one zone — not the full list-item record

Resource groups

- Workspace — page/cue grid (/apiv1/ws/info, /apiv1/ws/page)
- Thumbnails — cue and FX raster previews
- PangoScript — run commands and scripts; read/write documented objects
- Live Control — master, cue, and zone surfaces
- Zones — flag document on /apiv1/zone. UGC and preview writes use PangoScript.
- Playlist — transport control

Typical integration flow

- Discover — read workspace endpoints.
- Command — trigger cues or run scripts over REST.
- Poll — read live-control and playlist state (about once a second is enough).
- Host UI — serve custom HTML from the static Root, or run an external app on the LAN.

Guides

Build a cue grid UI

- GET /apiv1/ws/info — categories and page list (or /apiv1/ws/info2 for stable property names).
- GET /apiv1/ws/page?index=0 — cue names and empty slots.
- GET /apiv1/cuethumb?page=0&cue=0&size=150 — thumbnail for each slot. Add &time=2000 to render at 2 s.
- POST /apiv1/ps/execscript with Content-Type: text/plain and body CueDown 0,3 (or StartCue). One-line form: GET /apiv1/ps/cmd/cuedown%200,3 (%20 is a space; + is literal).

Control playlist playback

- GET /apiv1/playlist/list — available shows.
- GET /apiv1/playlist/play — start the current track. Use /apiv1/playlist/settrack?index=0 to jump first.
- GET /apiv1/playlist/state — position and transport. Poll this for a now-playing UI.
- GET /apiv1/playlist/setposition?time=120000 — seek in milliseconds.

Adjust live parameters

Read the master surface:

```
GET /apiv1/lc/master
```

Partial JSON write:

```
POST /apiv1/lc/master
Content-Type: application/json

{"scanrate": 80, "hue": 30}
```

The same pattern applies to /apiv1/lc/cue (per cue) and /apiv1/lc/zone?index=0 (per zone).

Read or adjust a projection zone

- GET /apiv1/psobject?name=zone — list (names, preview, uGC, 1-based projectorIndex).
- GET /apiv1/zone?index=0 — flags only (mute, blockzone, 0-based projectorindex, visualizationid, testframe, …). zone.0 is Live Control, not the full list item.
- POST /apiv1/zone?index=0 with {“visualizationid”:0,“testframe”:0}. ugc, mesh, name, and preview on this route return 400.
- Write UGC or preview with POST /apiv1/ps/execscript, for example: Zone.0.UGC.SizeX = 99
- Live parameters: GET/POST /apiv1/lc/zone?index=0.

Errors and script behavior

Code	Meaning
200	Success. Script calls return JSON with totalLines / executedLines / partial. No WriteLn text.
400	Validation or script error. Treat as failure.
404	Path is not under /apiv1/ and not in the static file root.
5xx	Internal BEYOND error. Check Quick Log on the show machine.

Script error body (1-based line):

```
{
  "error": "Line: 2, Error: Unknown identifier",
  "code": 400,
  "line": 2,
  "totalLines": 2,
  "executedLines": 1,
  "partial": true
}
```

Field validation example:

```
{
  "error": "Unknown or unsupported field",
  "field": "ugc"
}
```

- The whole execscript body is executed. There is no silent line cap.
- Execution stops at the first error. Earlier lines stay applied (partial: true).

- Only the first assignment on a semicolon-joined line runs.
- Out-of-range preview values may be stored and read back unclamped.
- GET /apiv1/ps/cmd/{command} is one statement. Query strings are stripped. %20 is a space. A literal + is not a space.

How the API behaves

- Read playlist or Live Control over HTTP whenever you need a fresh value.
- execscript returns a JSON result (status, line counts, partial). It does not return WriteLn / console text.
- /apiv1/zone is the zone flag document (mute, blockzone, projectorindex, and the other flag fields). Change UGC or preview geometry with execscript, for example Zone.0.UGC.SizeX = 99.
- psobject name=zone is the zone list. zone.<n> is that zone's Live Control properties.
- projectorindex on /apiv1/zone is 0-based. projectorIndex on the zone list is 1-based.

FAQ

Can I try API requests on this wiki page?

No. Use the API Playground on the BEYOND machine at http://<beyond-host>:8088/beyondswagger/.

Do REST calls require authentication?

Not in current builds.

Is HTTPS supported?

Not natively. Place a reverse proxy in front of BEYOND if you need TLS.

Why does my tablet get connection refused?

Use the BEYOND PC LAN IP, not 127.0.0.1. Allow inbound TCP on the HTTP port in Windows Firewall.

What Content-Type for scripts?

text/plain to POST /apiv1/ps/execscript. Do not send {"code":"…"}.

Where is the FX thumbnail endpoint?

GET /apiv1/fxthumb.

Are there rate limits?

None documented. Poll about once a second for UI feedback.

Which license includes the HTTP server?

BEYOND Ultimate only. See pangolin.com/collections/beyond-software.

How do I list or edit zones?

List: GET /apiv1/psobject?name=zone. Flags: GET/POST /apiv1/zone?index=0. UGC/preview: execscript.

Where is the machine-readable spec?

openapi.json in <exe>\www\beyondswagger, also attached beside this manual.

Endpoint reference

Every public REST path in OpenAPI 1.0. Replace <beyond-host> and the port with your show machine.

Method	Path	Summary
GET	/apiv1/ws/info	Workspace info – categories as keys
GET	/apiv1/ws/info2	Workspace info – fixed property names
GET	/apiv1/ws/page	Get one workspace page (info + cue array)
GET	/apiv1/cuethumb	Cue thumbnail (raster image)
GET	/apiv1/ps/cmd/{command}	Run a single PangoScript command (URI form)
POST	/apiv1/ps/execscript	Run a multi-line PangoScript
GET	/apiv1/psobject	Read a PangoScript object as JSON
POST	/apiv1/psobject	Write properties of a PangoScript object
GET	/apiv1/lc/master	Read Master Live Control
POST	/apiv1/lc/master	Write Master Live Control
GET	/apiv1/lc/cue	Read Cue Live Control
POST	/apiv1/lc/cue	Write Cue Live Control
GET	/apiv1/lc/zone	Read Projection Zone Live Control

Method	Path	Summary
POST	/apiv1/lc/zone	Write Projection Zone Live Control
GET	/apiv1/zone	Read projection zone flags
POST	/apiv1/zone	Write projection zone flags
GET	/apiv1/playlist/state	Get current playback state
GET	/apiv1/playlist/list	Get the show list and per-show metadata
GET	/apiv1/playlist/thumb	Get a playlist item thumbnail (raster image)
GET	/apiv1/playlist/play	Play
GET	/apiv1/playlist/stop	Stop
GET	/apiv1/playlist/first	First
GET	/apiv1/playlist/last	Last
GET	/apiv1/playlist/next	Next
GET	/apiv1/playlist/prev	Prev
GET	/apiv1/playlist/settrack	Jump to a specific track
GET	/apiv1/playlist/setposition	Seek within the current show
GET	/apiv1/fxthumb	FX-table thumbnail (raster image)

Workspace

GET /apiv1/ws/info

Workspace info – categories as keys

Returns the workspace structure: a `grid` block describing the page matrix, plus a `category` map whose keys are the user-defined category names in BEYOND (e.g. `Animals`, `Boxes`). The `All` key is always present and lists every page regardless of category.

Use this when you want a category-driven UI like the Cue Grid demo. If you need stable property names instead of category-name keys, use `/apiv1/ws/info2`.

Responses: 200

GET /apiv1/ws/info2

Workspace info – fixed property names

Same payload as `/apiv1/ws/info` but with stable property names instead of category names as keys. Easier to type-check; harder to drive a category-named UI.

Responses: 200

GET /apiv1/ws/page

Get one workspace page (info + cue array)

Returns the metadata for a single page (`pageName`, `pageCategory`) plus an array of all cues on that page. Cues are 0-indexed up to the workspace's `colCount * rowCount` (typically 64 = 8×8). The `empty` flag distinguishes blank slots from real cues so you can render the grid without flicker.

The page index is the linear page number across the whole workspace, not a per-category index. Get it from `/apiv1/ws/info`.

Parameters: index (query, required)

Responses: 200

Thumbnails

GET /apiv1/cuethumb

Cue thumbnail (raster image)

Returns a raster image preview of one workspace cue. Use this to populate cue grids without instantiating the cue. The server picks the closest pre-rendered size, then resamples; very small `size` values render fastest.

Tip: put `&time=2000` to render the cue at 2 s of its timeline rather than frame zero – useful for animated cues.

Parameters: page (query, required); cue (query, required); format (query, optional); size (query, optional); time (query, optional); backcolor (query, optional)

Responses: 200, 400

GET /apiv1/playlist/thumb

Get a playlist item thumbnail (raster image)

Returns a thumbnail for one show in the playlist. Unlike `/cuethumb` and `/fxthumb`, playlist thumbnails are user-supplied raster images baked into the show file at fixed sizes (see `thumb[]` in `/apiv1/playlist/list`). The server can't synthesize an exact size on demand – it picks the nearest pre-baked thumbnail to your requested `size`.

Provide either `size` or `image`:

- `size` – server returns the closest pre-baked size.

- `image` – server returns the Nth image stored in the show.

Parameters: index (query, required); size (query, optional); name (query, optional); image (query, optional)

Responses: 200

GET /apiv1/fxthumb

FX-table thumbnail (raster image)

Returns a raster preview of one effect from the FX table. Same parameter contract as `/apiv1/cuethumb` but indexed by `line` (0..7) and `index` (0..100) instead of page/cue.

On current BEYOND HTTP this path is `/apiv1/fxthumb`. The older root path `/fxthumb` returns 404.

Parameters: line (query, required); index (query, required); format (query, optional); size (query, optional); time (query, optional); backcolor (query, optional)

Responses: 200

PangoScript

GET /apiv1/ps/cmd/{command}

Run a single PangoScript command (URI form)

Everything after `/cmd/` is one PangoScript statement.

Query parameters are stripped and never become script. The path is percent-decoded once (`%20` is a space). A literal + is not a space.

For multi-line scripts, or any statement with spaces, use `POST /apiv1/ps/execscript` with `Content-Type: text/plain`.

Examples

- `GET /apiv1/ps/cmd/blackout`
- `GET /apiv1/ps/cmd/master.brightness=75`
- `GET /apiv1/ps/cmd/cuedown%200,3`
- `GET /apiv1/ps/cmd/Zone.0.Preview.PositionX%20=%200`
- `GET /apiv1/ps/cmd/cuedown+0,3` — works because `+0` is the number zero, not a space

Parameters: command (path, required)

Responses: 200, 400

POST /apiv1/ps/execscript

Run a multi-line PangoScript

Body is UTF-8 PangoScript source (`Content-Type: text/plain`). Newlines separate statements. JSON wrappers such as `{"code":"..."}` are not supported.

The whole body is executed. There is no silent line cap. Execution stops at the first error; earlier lines stay applied (`partial: true`). Only the first assignment on a `;`-joined line runs. Out-of-range preview values may be stored unclamped.

`WriteLn` / echo text is not returned. Treat any non-2xx status as failure.

Request body: text/plain script

Responses: 200, 400

```
Master.Brightness=75
Master.Hue=20
```

GET /apiv1/psobject

Read a PangoScript object as JSON

Returns the live JSON state of a documented object: `master`, `playlist`, `cue.<page>.<index>`, `zone` (list), or `zone.<n>` (Live Control subset).

`zone` list items include name, preview, uGC, and flags. `projectorIndex` on the list is 1-based. Top-level `count` appears once. `zone.<n>` is not the same shape as a list item.

Parameters: name (query, required)

Responses: 200, 400, 404

POST /apiv1/psobject

Write properties of a PangoScript object

Partial JSON update of a documented writable object. `zone.<n>` accepts Live Control fields (for example `posX`). `name` and `preview` on `zone.<n>` return `400`. Unknown properties return `400`.

Parameters: name (query, required)

Request body: application/json

Responses: 200, 400, 404

```
{
  "BPM": 240
}
```

Live Control

GET /apiv1/lc/master

Read Master Live Control

Returns the current state of the Master Live Control surface – the global brightness/hue/scanrate/etc. layer that overlays everything. Equivalent to reading `master` via `/apiv1/psobject` but returns only the Live Control subset of properties.

For a per-cue Live Control, see `/apiv1/lc/cue`. For a per-zone Live Control, see `/apiv1/lc/zone`.

Responses: 200

POST /apiv1/lc/master

Write Master Live Control

Send a partial Live Control document; only the keys you include are mutated. To reset a property to default, omit it (POSTing `null` does not reset).

Request body: application/json

Responses: 200, 400

```
{
  "scanrate": 80,
  "hue": 30
}
```

GET /apiv1/lc/cue

Read Cue Live Control

Per-cue Live Control. The Live Control values on a cue are the *current* values, not the design-time defaults – they reflect any tweaks made while the cue is playing.

Parameters: page (query, required); index (query, required)

Responses: 200

POST /apiv1/lc/cue

Write Cue Live Control

Parameters: page (query, required); index (query, required)

Request body: application/json

Responses: 200

GET /apiv1/lc/zone

Read Projection Zone Live Control

Per-zone Live Control surface. Same property set as `/apiv1/lc/master` but applied to one projection zone.

Parameters: index (query, required)

Responses: 200

POST /apiv1/lc/zone

Write Projection Zone Live Control

Parameters: index (query, required)

Request body: application/json

Responses: 200

Zones

GET /apiv1/zone

Read projection zone flags

Returns the flag document for one zone: `mute`, `blockzone`, `projectorindex`, `active`, `visualizationid`, `ratiox`, `ratioy`, `testframe`, `visible`.

`projectorindex` is 0-based. This route does not return `ugc`, `mesh`, `name`, or `preview`. For the full list record (including uGC and preview), use `GET /apiv1/psobject?name=zone`.

Parameters: index (query, required)

Responses: 200

POST /apiv1/zone

Write projection zone flags

Send a partial flag document. Supported writes include `visualizationid`, `testframe`, and `projectorindex` (0-based integer in engine range).

`name`, `preview`, `ugc`, and `mesh` are not writable here — they return `400` (`Unknown or unsupported field`) and the zone is left unchanged. Non-JSON bodies and unknown fields also return `400` before any write.

To write UGC or preview geometry, use `POST /apiv1/ps/execscript`.

Parameters: index (query, required)

Request body: application/json

Responses: 200, 400

```
{
  "visualizationid": 0,
  "testframe": 0
}
```

Playlist

GET /apiv1/playlist/state

Get current playback state

Cheap snapshot of the playlist transport: whether it's playing, the position in milliseconds, the duration in milliseconds, and the current track index. Most other Playlist endpoints also return this object so you can update your UI from a single response shape.

Responses: 200

GET /apiv1/playlist/list

Get the show list and per-show metadata

Returns the complete playlist plus per-show metadata: name, comment, file paths, audio file, duration, timecode, and a list of pre-rendered thumbnail sizes. Use it to populate a playlist UI.

Responses: 200

GET /apiv1/playlist/play

Play

Start playback of the current track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/stop

Stop

Stop playback. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/first

First

Jump to the first track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/last

Last

Jump to the last track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/next

Next

Advance to the next track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/prev

Prev

Step back to the previous track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/settrack

Jump to a specific track

Selects a track by zero-based index. Equivalent to clicking a row in the playlist UI. Returns the updated Playback Status.

Parameters: index (query, required)

Responses: 200

GET /apiv1/playlist/setposition

Seek within the current show

Sets the playback head, in milliseconds from the start of the current show. Use the `infoDurationMS` field from `/apiv1/playlist/list` for the upper bound. Returns the updated Playback Status.

Parameters: time (query, required)

Responses: 200

Zone flag fields

GET/POST /apiv1/zone returns and accepts only these fields:

Field	Type	Notes
mute	integer	
blockzone	integer	
projectorindex	integer	0-based projector slot. Out of range → 400, zone unchanged.
active	integer	
visualizationid	integer	Writable
ratiox	number	
ratioy	number	
testframe	integer	Writable

Field	Type	Notes
visible	integer	

PangoScript command catalog

529 commands from the https://wiki.pangolin.com/pangoscript-command-list.html. Run them with POST /apiv1/ps/execscript (text/plain) or GET /apiv1/ps/cmd/{{command}} for a single statement.

Import into other tools

Attach or download openapi.json (OpenAPI 3.0.3) and import it into Postman, Insomnia, Bruno, or a code generator. The same file ships in <exe>\www\beyondswagger\openapi.json.

Example pages (cue grid, playlist, live control) ship in the installer www folder.

Common mistakes

- Using 127.0.0.1 from another device. Use the BEYOND PC LAN IP.
- Calling paths without the /apiv1/ prefix.
- Sending {“code”:“…”} to execscript instead of raw text/plain.
- Treating + as a space on ps/cmd. Use %20.
- POSTing ugc or preview to /apiv1/zone — that returns 400. Use execscript.
- Leaving static Root blank but expecting files next to BEYOND.exe instead of in <exe>\www. The 5.8 installer fills www.

From:
http://wiki.pangolin.com/ - **Complete Help Docs**

Permanent link:
http://wiki.pangolin.com/doku.php?id=beyond:http_server

Last update: **2026/10/24 10:06**