



Outdoor Lighting Controller

Pixout Controller **API reference**, version 1.3

PIXOUT
media server

Changes

Data	Description
26.04.2018	Added Change cuelist speed article, 2.4

Introduction

Pixout controller software based on RESTfull web server.

The following document describes the RESTful API of the Pixout controller.

For configuration needs you can use User or Admin web panel.

User/Admin panel interface is written as modern web application and could be accessibly from any modern web browser or iOS application (PixoutRemote).

The controller has two network interfaces

1. Static wire interface 2.0.0.230 for ArtNet recording/replaying
<http://2.0.0.230>
2. Dynamic (DHCP) WiFi zero config interface “pixoutserver.local” for control purpose
<http://pixoutserver.local>

Due to heavy load during ArtNet recording/replaying it's highly not recommended to use static IP address for device control purpose. Static IP designed for Art-Net recording/replaying or initial device configuration only.

1 General

1.1 Get system status

Retrieve system status from the controller

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/status
HTTP	GET

Response codes:

200 – Status received successfully

500 – Can't read status information

Response data format:

application/json

Response entities:

1. PID – In-use cuelist ID number
2. OP – Operation in progress (play or stop)

3. Fps – Speed (Frames per second)
4. Intensity – Intensity for current cue list
5. UpdateUID – Informs when cue lists changed and you need to update it (if value changed)

Response entities for errors:

1. Code – Error code
2. Message – Error message

Example:

HTTP GET → http://pixoutserver.local/px/v1/player/status

STATUS: 200

Data:

```
{"PID":0,"OP":"play","Fps":0,"Intensity":100,"UpdateUID":134049602}
```

HTTP GET → http://pixoutserver.local/px/v1/player/status

STATUS: 500

Data in “application/json” format:

```
{"Code":400,"Message":"Can't read grabber status"}
```

2 Cuelists

2.1 Get cuelists

Get list of existing cuelists

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/playlists
HTTP	GET

Response codes:

200 – List received successfully

500 – Can't read cuelist, incorrect json syntax

Response data format:

application/json

Response entities:

1. Id – Cuelist ID number
2. Name – Cuelist name
3. Items – Amount of cues inside cuelist
4. Icon – Cuelist image (full path: http://pixoutserver.local/images/pl_icons/{Icon})

Note: Path is deprecated parameter and should not be used any more

Response entities for errors:

1. Code – Error code
2. Message – Error message

Example:

HTTP GET → <http://pixoutserver.local/px/v1/playlists>

STATUS: 200

Data:

```
[{"Id":0,"Name":"Demo","Items":1,"Icon":"relax.jpg","Path":"playlists/0.json"},
{"Id":1,"Name":"White stars","Items":1,"Icon":"white-stars.jpg","Path":"playlists/1.json"},
{"Id":2,"Name":"Green forest","Items":1,"Icon":"green-forest.jpg","Path":"playlists/2.json"},
{"Id":3,"Name":"Transitions","Items":1,"Icon":"nice_trans.jpg","Path":"playlists/3.json"}]
```

2.2 Play/Pause/Stop cueлист

Play or pause specified cueлист

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/pl/{cueлист}/play http://pixoutserver.local/px/v1/player/pl/{cueлист}/pause http://pixoutserver.local/px/v1/player/pl/{cueлист}/stop
Cueлист	Cueлист ID number
HTTP	PUT

Response codes:

200 – Command executed successfully

400 – Incorrect cueлист ID

400 – Can't proceed play/pause/stop

Response data format:

application/json

Response entities for errors:

1. Code – Error code
2. Message – Error message

Example:

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/0/play>

STATUS: 200

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/999/play>

STATUS: 400

Data in “application/json” format:

```
{"Code":400,"Message":"Play, Incorrect cueлист ID"}
```

2.3 Change cuelist intensity

Change brightness for specified cuelist.

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/pl/{cuelist}/setIntensity/{intsy}
Cuelist	Cuelist ID number
Intsy	Intensity number from 0 till 100 (note: 0 - mean black out)
HTTP	PUT

Response codes:

200 – Brightness changed successfully

400 – Incorrect cuelist ID

400 – Render not found

404 – Internal errors

Response entities for errors:

1. Code – Error code
2. Message – Error message

Example:

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/0/setIntensity/100>

STATUS: 200

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/999/setIntensity/100>

STATUS: 400

Data in “application/json” format:

```
{"Code":400,"Message":"Intensity, Render not found, can proceed"}
```

2.4 Change cuelist speed

Change speed for specified cuelist.

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/pl/{cuelist}/setFPS/{speed}
Cuelist	Cuelist ID number
Speed	Speed number from 0 till 100 (note: 0 - as it was recorded)
HTTP	PUT

Response codes:

200 – Speed changed successfully

400 – Incorrect cuelist ID

400 – Render not found

404 – Internal errors

Response entities for errors:

3. Code – Error code

4. Message – Error message

Example:

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/0/setFPS/100>

STATUS: 200

HTTP PUT → <http://pixoutserver.local/px/v1/player/pl/999/setFPS/100>

STATUS: 400

Data in “application/json” format:

```
{ "Code":400,"Message":"Speed, Incorrect cuelist ID" }
```

2.5 Get Cuelist

Get cuelist

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/pl/{cuelist}
Cuelist	Cuelist ID number
HTTP	GET

Response codes:

200 – Cuelist received successfully

400 – Incorrect cuelist ID syntax

404 – Incorrect cuelist ID

500 – Internal errors

Response entities for errors:

5. *Code* – Error code

6. *Message* – Error message

Example:

HTTP GET → <http://pixoutserver.local/px/v1/player/pl/0>

STATUS: 200

Data in “application/json” format:

```
[{"Id":0,"path":"effects/0/demo_effect","fps":30,"timeout":60000,"Icon":"relax.jpg","name":"Demo", "outputs":[]}]
```

HTTP GET → <http://pixoutserver.local/px/v1/player/pl/999>

STATUS: 404

Data in “application/json” format:

```
{"Code":404,"Message":"Cuelist ID not found"}
```

2.6 Update Cuelist

Update cuelist

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/player/pl/{cuelist}
Cuelist	Cuelist ID number
Data	Post data in json format: [{"Id":0,"path":"effects/0/demo_effect","fps":30,"timeout":60000,"Icon":"relax.jpg","name":"Demo","outputs":[]}]
HTTP	POST

Response codes:

- 200 – Cuelist updated successfully
- 400 – Incorrect cuelist ID syntax
- 404 – Incorrect cuelist ID
- 500 – Internal errors

Response entities for errors:

- 7. Code – Error code
- 8. Message – Error message

Example:

HTTP POST → <http://pixoutserver.local/px/v1/player/pl/0>

“`[{"Id":0,"path":"effects/0/demo_effect","fps":30,"timeout":60000,"Icon":"relax.jpg","name":"Demo","outputs":[]}]`”

STATUS: 200

HTTP POST → <http://pixoutserver.local/px/v1/player/pl/999>

STATUS: 404

Data in “application/json” format:

<code>{"Code":404,"Message":"Cuelist ID not found"}</code>
--

3 Grabbing

3.1 Start grabbing

Start Art-Net data grabbing.

Attention: This endpoint can record data but if you want to insert newly recorded cue into the cuelist you need to do it independently.

First of all get specified cue (Get Cuelist endpoint) list and find maximum Cue ID number. Than insert new json record with correct cue path and call update cuelist endpoint.

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/grabber/pl/{cuelist}/start/{name}
Cuelist	Cuelist ID number
Name	Recorded cue name (allowed characters “a-Z”, “A-Z”, “0-9”, “_”, “-”)
HTTP	PUT

Response codes:

- 200 – Grabbing started successfully
- 400 – Incorrect cuelist syntax
- 400 – Render not found
- 404 – Not existing cuelist ID
- 500 – Internal error

Response entities for errors:

- 9. Code – Error code
- 10. Message – Error message

Example:

HTTP PUT → http://pixoutserver.local/px/v1/grabber/pl/0/start/test_cue-1

STATUS: 200

HTTP GET → <http://pixoutserver.local/px/v1/grabber/pl/0>

Data in “application/json” format:

```
[{"Id":0,"path":"effects/0/demo_effect","fps":30,"timeout":60000,"Icon":"relax.jpg","name":"Demo","outputs":[]}]
```

STATUS: 200

HTTP POST → <http://pixoutserver.local/px/v1/grabber/pl/0>

Data in “application/json” format:

```
[{"Id":0,"path":"effects/0/demo_effect","fps":30,"timeout":60000,"Icon":"relax.jpg","name":"Demo", "outputs":[]}, {"Id":1,"path":"effects/0/test_cue-1", "fps":30,"timeout":60000,"name":"New Demo", "outputs":[]}]
```

HTTP PUT → http://pixoutserver.local/px/v1/grabber/pl/999/start/test_cue-1

STATUS: 404

Data in “application/json” format:

```
{"Code":404,"Message":"Grabber, Cuelist ID didn't found"}
```

Note:

This end-point syntax could be changed in future API version

3.2 Stop grabbing

Stop Art-Net data grabbing

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/grabber/stop
HTTP	PUT

Response codes:

200 – Grabbing stopped successfully

500 – Internal error

Response entities for errors:

11. Code – Error code

12. Message – Error message

Example:

HTTP PUT → <http://pixoutserver.local/px/v1/grabber/stop>

STATUS: 200

HTTP PUT → <http://pixoutserver.local/px/v1/grabber/stop>

STATUS: 500

Data in “application/json” format:

```
{ "Code":500,"Message":"Grabber, Can't stop grabber service" }
```

4 Get Status

Get grabbing status

Authentication: not required

Request:

NAME	DESCRIPTION
URI	http://pixoutserver.local/px/v1/grabber/status
HTTP	GET

Response codes:

200 – Grabbing status received successfully

500 – Internal error

Response data format:

application/json

Response entities for errors:

13. Code – Error code

14. Message – Error message

Example:

HTTP GET → <http://pixoutserver.local/px/v1/grabber/status>

STATUS: 200

Data in “application/json” format:

```
{"status": "play"}
```

Status codes: “play”, “stop”, “pause”, “error: {code}”

HTTP GET → <http://pixoutserver.local/px/v1/grabber/status>

STATUS: 500

Data in “application/json” format:

```
{"Code": 500, "Message": "Grabber, Can't stop grabber service"}
```