

RetailPlayer Remote Control REST API

This REST API is used to monitor and control RetailPlayer devices. The core blocks of this API are:

GET requests used to retrieve parameters:

- Status
- Current Volume
- Mute / Un-mute Status
- Channel part of the channel list assigned to the device on the RetailPlayer Portal
- Cues available and already assigned to the device on the RetailPlayer Portal

PUT requests used to set parameters:

- Volume of the device
- Mute / Un-mute
- Change channel among the ones available in the channel list selected in the Portal

POST requests are also used to:

- Trigger playback of an asset part of the list of cues assigned to the device
- Stop any currently playing asset

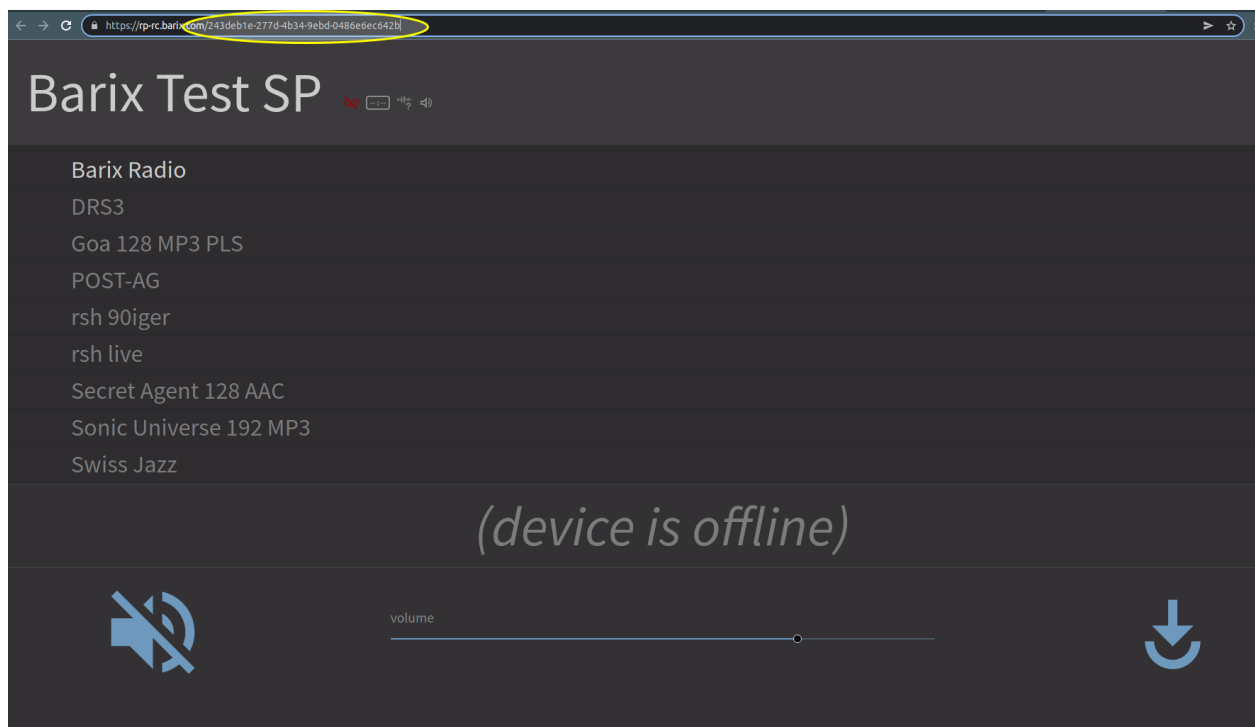
Table of Contents

Access the API	3
API transport and path	4
Authorization	4
Methods	5
GET /status	5
GET /volume	8
GET /muted	8
GET /channels	9
GET /triggers	10
PUT /volume	12
PUT /muted	12
PUT /channels	13
POST /triggers	14

Access the API:

To access the API are required 2 items:

1. The customer **ApiKey**: provided privately from Barix via email.
2. The **device-control uuid**: is the single device identifier. Each call must be addressed to a specific RetailPlayer device. This is what identifies the single device. The device-control UUID is available in the URL a QR-Code generated for a device. Once the QR-Code is generated, open the link into a web browser, the last portion of the path is the device-control UUID (an example is this: "ec46854b-82c7-43f1-a534-5e1f1w6r253a") - The below image clearly shows where to retrieve the device-control UUID.



API transport and path:

The transport must be over HTTPS.

The host where to send the calls is: <https://manage.barix.com/>

Refer to each method path to know the path for the individual requests.

There is a rate limit of 50 calls per second that prevents overloads on the RetailPlayer Portal side. When using the API as part of your program make sure to do not exceed this limit and use a delay of at least 20ms between each call sent, otherwise you will get *Error 503 - Service unavailable* responses.

Authorization

The calls must be authorized using the ApiKey provided by Barix. It has to be sent in the **http header** as follow:

Key	x-retailplayer-rc-apikey
Value	<apiKey>

Methods

GET /status

Allows to retrieve the status information sent by the device to the Portal.

Method	GET
Path	/rest/v1/device-control/< device-control-uuid >/status
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	NONE
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	<pre>{ "online": bool, "volume": int, "muted": bool, "channel": "string", "url": "string", "fallbackUrl": "string", "activeResource": "string", "activeStream": "string", "scheduleStatus": "string", "upTime": int, "firmwareVersion": "string", "bufferLevel": "string", "channelListName": "string" }</pre>

Description and examples of the response body of the /status call

key	values	description
online	true	Device is currently connected to the Portal
	false	Device is currently NOT connected with the Portal
volume	min: 0 max: 100	current volume of the device in the range of 0-100
muted	true	Device is currently muted
	false	Device is currently NOT muted
channel	"channelName"	The name of the currently assigned channel
url	"http://myradio.com:4567/radio1"	The Main URL of the channel currently assigned to the device
fallbackUrl	"http://myradio.com:4567/radio2"	The Fallback URL of the channel currently assigned to the device
activeResource	"MAIN"	When the device plays the Main URL
	"FALLBACK"	When the device plays the Fallback URL
	"FAILOVER TRACKS"	When the device plays the failover tracks
	"EVENT"	When the device is playing an event (a scheduled asset or a cue)
	"PRIORITY"	When the device plays the priority URL
	"USB"	When the device plays from the connected USB stick
	"NONE"	When the device is not playing anything

activeStream	"http://myradio.com:4567/radio1"	The name of the currently playing resource. In case of a file playing is the name of the file.
scheduleStatus	"Active"	Shows the status of the Opening Hours Scheduler. Active means the device is in "active" mode.
	"Inactive"	Shows the status of the Opening Hours Scheduler. Inactive means the device is in "inactive" mode.
upTime	1234	The amount of time the device is being powered in minutes.
firmwareVersion	"6.5.0"	The firmware version the device is currently running.
bufferLevel	min: 0 max: 60	The amount of buffer filled in seconds. It is a number between 0 and 60, since the buffer could keep max. 1 min worth of audio samples.
channelListName	"My Channel List"	The name of the channel list currently assigned to the device

GET /volume

Allows to retrieve the currently set volume of the device in the moment the call is being sent.

Method	GET
Path	/rest/v1/device-control/<device-control-uuid>/volume
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<i>NONE</i>
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	<pre>{ "value": int }</pre>

It returns the value of the currently set volume. It could be a number between 0 and 100.

GET /muted

Allows to retrieve the information about the mute status of the device

Method	GET
Path	/rest/v1/device-control/<device-control-uuid>/muted
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<i>NONE</i>
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	<pre>{ "value": bool }</pre>

It returns **true** if the device is muted, **false** otherwise.

GET /channels

Allows to retrieve the channel catalog part of the channel list assigned to the device in the moment the call is being sent.

Method	GET
Path	/rest/v1/device-control/ <device-control-uuid> /channels
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	NONE
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	<pre>{ "value": [{ "id": "string", "name": "string", "url": "string", "fallbackUrl": "string" }] }</pre>

It returns an array of objects where each object is a channel entity part of the channel list currently assigned to the device. For each channel it is reported:

key	values	description
id	"4b5cce21-2530-4951-b74d-9b6baa8568e1"	UUID of the corresponding channel
name	"my radio"	The name assigned to the channel
url	"http://myradio.com:4567/radio1"	The MAIN URL configured for this channel
fallbackUrl	"http://myradio.com:4567/radio2"	The FALLBACK URL configured for this channel

GET /triggers

Allows to retrieve the information about the currently assigned CUE SET (triggered files)

Method	GET
Path	/rest/v1/device-control/<device-control-uuid>/triggers
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	NONE
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	<pre>{ "value": [{ "id": "string", "name": "string", "ordinal": int, "asset": { "id": "string", "name": "string", "mime": "string", "tags": { "duration": "string" }, "thumbnail": "string (base64)" } }] }</pre>

It returns an array of objects where each object contains information about the single cuing buttons and the assets associated with it.

key	values	description
id	"4b5cce21-2530-4951-b74d-9b6baa8568e1"	UUID of the corresponding cue
name	"Happy Birthday"	The name assigned to the cue
ordinal	1	The position of the cue in the order assigned to the cue set configuration

asset	object	it contains the information about the asset (audio file) associated with the corresponding cue. See table below with details on the Asset
--------------	--------	---

Asset object details

key	values	description
id	"4b5cce21-2530-4951-b74d-9b6baa8568e1"	UUID of the corresponding asset
name	"HappyBirthday.mp3"	The name assigned to the cue
mime	"audio/x-mp3"	The mime type of the asset
tags / duration	"4.899"	A tag specifying the duration of the asset in a string in seconds
thumbnail	base64 string	If an image is associated with the asset it is provided in a base64 string as it is extracted by the file itself

PUT /volume

Allows to set the volume of a RetailPlayer device

Method	PUT
Path	/rest/v1/device-control/<device-control-uuid>/volume
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<pre>{ "value" : int }</pre>
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	

The value passed in the body must be an integer between 0 and 100.

PUT /muted

Allows to mute / un-mute of a RetailPlayer device

Method	PUT
Path	/rest/v1/device-control/<device-control-uuid>/muted
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<pre>{ "value" : bool }</pre>
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	

The value passed in the body must be either **true**, to mute the device, **false**, to unmute

PUT /channels

Allows to change a channel on a RetailPlayer Device

Method	PUT
Path	/rest/v1/device-control/ <device-control-uuid> /channels
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<pre>{ "value" : "string" }</pre>
Example	<pre>{ "value" : "4b5cce21-2530-4951-b74d-9b6baa8568e1" }</pre>
Response Status	200, 400, 401, 404, 409, 500
Response Headers	content-type: application/json
Response Body	

The value passed in the body must correspond to the **UUID** one of the channels present in the channel catalog of the channel list currently assigned to the device, if the uuid is not in the channel catalog the Portal will complain in the response status 409.

POST /triggers

Allows to play an asset from the cue set currently assigned to the device. It also allows to STOP any currently playing asset.

Method	POST
Path	/rest/v1/device-control/ <device-control-uuid> /triggers
Request Headers	x-retailplayer-rc-apikey: <apiKey>
Request Body	<pre>{ "action": "string", "value" : "string" }</pre>
Example	<pre>{ "action": "PLAY", "value" : "4b5cce21-2530-4951-b74d-9b6baa8568e1" }</pre>
Response Status	200, 400, 401, 404, 500
Response Headers	content-type: application/json
Response Body	

Explanation of the parameters of this call:

key	values	description
action	"PLAY"	Instructs the device to play the asset specified in the value parameter
	"STOP"	Instructs the device to stop playing any asset currently playing. When STOP is being sent there is no need to pass the value in the body of the call.
value	"4b5cce21-2530-4951-b74d-9b6baa8568e1"	The UUID of the cue that has to PLAY. not needed when "action": "STOP" is sent.