

Obtaining stream information from backend server

- [Current streams information](#)
 - [Current instantaneous stream metric values](#)
- [Stream history](#)

Information about the streams on monitored nodes can be obtained from backend server with [Websocket API](#) or [REST API](#).

Current streams information

Current streams information can be obtained with /api/stream/list request

API	Request	Response	Response status
WS API	<pre>SEND destination:/app/api/stream/list content-length:207 { "requestId":"f2378c35-e7ca-4241-b382-29f696f0dddb", "realm":"/api/stream/list", "payload":{ { "node":"", "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "name":"", "publisher":"", "status":"", "showClosed":false, "nodeIp":"192.168.1.5", "nodeHostName":"" } } }</pre>	<pre>MESSAGE destination:/user/service content-type:application/json;charset=UTF-8 subscription:sub-1 message-id:3-35 content-length:210 { "requestId":"f2378c35-e7ca-4241-b382-29f696f0dddb", "status":200, "reason":"SUCCESS", "payload":[{ "name":"test", "publisher":null, "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "age":229, "status":"PUBLISHING" }] }</pre>	200 OK 400 Object not found 500 Persist exception
REST API	<pre>POST: /api/stream/list "application/json; charset=utf-8" { "node":"", "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "name":"", "publisher":"", "status":"", "showClosed":false, "nodeIp":"", "nodeHostName":"yourserver.com" }</pre>	<pre>{ "status": 200, "reason": "SUCCESS", "payload": [{ "name": "test", "publisher": null, "mediaId": "f3ad4670-8e83-11e9-8386-dbc3d191a79a", "age": 152, "status": "PUBLISHING" }] }</pre>	200 OK 400 Object not found 500 Persist exception

Where

- node – node identifier (mandatory)
- mediaId – media session identifier
- name – stream name
- age – stream cache record age in milliseconds
- publisher - publisher identifier (for played streams)
- status – stream status (PENDING, PUBLISHING, PLAYING, UNPUBLISHED, STOPPED, FAILED)
- showClosed - show closed streams from cache
- nodeIp - IP address of server monitored
- nodeHostName - hostname of server monitored

Response data are obtained from current metrics cache.

Current instantaneous stream metric values

Current instantaneous metric values for stream published or played on WCS can be obtained with /api/stream/metrics request

API	Request	Response	Response status
WS API	SEND destination:/app/api/stream/metrics content-length:176 { "requestId":"e6d022e5-b5ae-4b6e-ad6e-e1c188e324d7", "realm":"/api/stream/metrics", "payload": { "node":"3", "mediaId":"5d03b340-8e8c-11e9-92b7-23d635edebe6", "limit":10, "offset":0 } } }	MESSAGE destination:/user/service content-type:application/json;charset=UTF-8 subscription:sub-1 message-id:2-18 content-length:1764 { "requestId":"e6d022e5-b5ae-4b6e-ad6e-e1c188e324d7", "status": 200, "reason": "SUCCESS", "payload": [{ "AUDIO_SYNC": 3769496857356, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27:37.644Z", "AUDIO_RATE": 29816, "seq": 14, "AUDIO_LOST": 41 }, { "AUDIO_SYNC": 3769496857236, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27:37.52Z", "AUDIO_RATE": 31048, "seq": 14, "AUDIO_LOST": 41 }, { "VIDEO_SYNC": 3769496857221, "VIDEO_CODEC": 119, "VIDEO_NACK": 38, "VIDEO_PLI": 0, "VIDEO_RATE": 404488, "VIDEO_WIDTH": 320, "time": "2019-06-14T10:27:37.494Z", "VIDEO_FPS": 23, "VIDEO_HEIGHT": 240, "seq": 16 }, { "AUDIO_SYNC": 3769496857116, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27:37.397Z", "AUDIO_RATE": 29304, "seq": 14, "AUDIO_LOST": 41 }, { "VIDEO_SYNC": 3769496857029, "VIDEO_CODEC": 119, "VIDEO_NACK": 38, "VIDEO_PLI": 0, "VIDEO_RATE": 409056, 	200 OK 400 Object not found 500 Persist exception

		<pre> "VIDEO_WIDTH": 320, "time": "2019-06-14T10:27: 37.301Z", "VIDEO_FPS": 22, "VIDEO_HEIGHT": 240, "seq": 16 }, { "AUDIO_SYNC": 3769496856996, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 37.285Z", "AUDIO_RATE": 28112, "seq": 14, "AUDIO_LOST": 41 }, { "AUDIO_SYNC": 3769496856876, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 37.16Z", "AUDIO_RATE": 28248, "seq": 14, "AUDIO_LOST": 41 }, { "VIDEO_SYNC": 3769496856806, "VIDEO_CODEC": 119, "VIDEO_NACK": 37, "VIDEO_PLI": 0, "VIDEO_RATE": 430992, "VIDEO_WIDTH": 320, "time": "2019-06-14T10:27: 37.124Z", "VIDEO_FPS": 22, "VIDEO_HEIGHT": 240, "seq": 16 }, { "AUDIO_SYNC": 3769496856756, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 37.039Z", "AUDIO_RATE": 27424, "seq": 14, "AUDIO_LOST": 41 }, { "AUDIO_SYNC": 3769496856636, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 36.923Z", "AUDIO_RATE": 26888, "seq": 14, "AUDIO_LOST": 41 } }] } </pre>	
REST API		<pre> { "status": 200, "reason": "SUCCESS", "payload": [{ "AUDIO_SYNC": 3769496857356, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 37.644Z", </pre>	200 OK 400 Object not found 500 Persist exception

```
POST: /api/stream/metrics "application/json; charset=utf-8"
{
  "node": "3",
  "mediaId": "5d03b340-8e8c-11e9-92b7-23d635edebe6",
  "limit": 10,
  "offset": 0
}
```

```

    "AUDIO_RATE": 29816,
    "seq": 14,
    "AUDIO_LOST": 41
  },
  {
    "AUDIO_SYNC": 3769496857236,
    "AUDIO_CODEC": 111,
    "time": "2019-06-14T10:27:
37.52Z",
    "AUDIO_RATE": 31048,
    "seq": 14,
    "AUDIO_LOST": 41
  },
  {
    "VIDEO_SYNC": 3769496857221,
    "VIDEO_CODEC": 119,
    "VIDEO_NACK": 38,
    "VIDEO_PLI": 0,
    "VIDEO_RATE": 404488,
    "VIDEO_WIDTH": 320,
    "time": "2019-06-14T10:27:
37.494Z",
    "VIDEO_FPS": 23,
    "VIDEO_HEIGHT": 240,
    "seq": 16
  },
  {
    "AUDIO_SYNC": 3769496857116,
    "AUDIO_CODEC": 111,
    "time": "2019-06-14T10:27:
37.397Z",
    "AUDIO_RATE": 29304,
    "seq": 14,
    "AUDIO_LOST": 41
  },
  {
    "VIDEO_SYNC": 3769496857029,
    "VIDEO_CODEC": 119,
    "VIDEO_NACK": 38,
    "VIDEO_PLI": 0,
    "VIDEO_RATE": 409056,
    "VIDEO_WIDTH": 320,
    "time": "2019-06-14T10:27:
37.301Z",
    "VIDEO_FPS": 22,
    "VIDEO_HEIGHT": 240,
    "seq": 16
  },
  {
    "AUDIO_SYNC": 3769496856996,
    "AUDIO_CODEC": 111,
    "time": "2019-06-14T10:27:
37.285Z",
    "AUDIO_RATE": 28112,
    "seq": 14,
    "AUDIO_LOST": 41
  },
  {
    "AUDIO_SYNC": 3769496856876,
    "AUDIO_CODEC": 111,
    "time": "2019-06-14T10:27:
37.16Z",
    "AUDIO_RATE": 28248,
    "seq": 14,
    "AUDIO_LOST": 41
  },
  {
    "VIDEO_SYNC": 3769496856806,
    "VIDEO_CODEC": 119,
    "VIDEO_NACK": 37,
    "VIDEO_PLI": 0,
```

		<pre> "VIDEO_RATE": 430992, "VIDEO_WIDTH": 320, "time": "2019-06-14T10:27: 37.124Z", "VIDEO_FPS": 22, "VIDEO_HEIGHT": 240, "seq": 16 }, { "AUDIO_SYNC": 3769496856756, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 37.039Z", "AUDIO_RATE": 27424, "seq": 14, "AUDIO_LOST": 41 }, { "AUDIO_SYNC": 3769496856636, "AUDIO_CODEC": 111, "time": "2019-06-14T10:27: 36.923Z", "AUDIO_RATE": 26888, "seq": 14, "AUDIO_LOST": 41 }] } }</pre>	
--	--	---	--

Здесь:

- node – node identifier (mandatory)
- mediaId – media session identifier (mandatory)
- limit - messages quantity in cache selection
- offset - cache offset
- time - time stamp
- seq - cache message sequence number

Response also contains current instantaneous metric values from cache.

Stream history

Stream history information can be obtained with /api/stream/history request

API	Request	Response	Response status
-----	---------	----------	-----------------

WS API	<pre> SEND destination:/app/api/stream/history content-length:191 { "requestId":"4a700886-a559-42b7-b43d-82679d8573d6", "realm":"/api/stream/history", "payload": { "node":"3", "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "name":"", "publisher":"", "status":"", "limit":1, "offset":0 } } </pre>	<pre> MESSAGE destination:/user/service content-type:application/json;charset=UTF-8 subscription:sub-1 message-id:3-39 content-length:310 { "requestId":"4a700886-a559-42b7-b43d-82679d8573d6", "status":200, "reason":"SUCCESS", "payload":[{ "id":265, "name":"test", "status":"UNPUBLISHED", "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "createDate":"2019-06-14T09:08:20.560+0000", "endDate":null, "publisher":0, "duration":598658, "subscribers":[], "node":3 }] } </pre>	200 OK 400 Object not found 500 Persist exception
REST API	<pre> POST: /api/stream/history "application/json; charset=utf-8" { "node":"3", "mediaId":"f3ad4670-8e83-11e9-8386-dbc3d191a79a", "name":"", "publisher":"", "status":"", "limit":1, "offset":0 } </pre>	<pre> { "status": 200, "reason": "SUCCESS", "payload": [{ "id": 265, "name": "test", "status": "UNPUBLISHED", "mediaId": "f3ad4670-8e83-11e9-8386-dbc3d191a79a", "createDate": 1560503300560, "endDate": null, "publisher": 0, "duration": 598658, "subscribers": [], "node": 3 }] } </pre>	200 OK 400 Object not found 500 Persist exception

Where

- node – node identifier
- mediaId – media session identifier
- name – stream name
- publisher – publisher identifier (for played streams)
- status – stream status (PENDING, PUBLISHING, PLAYING, UNPUBLISHED, STOPPED, FAILED)
- limit – maximum stream count in DB selection
- offset – offset from stream table beginning
- id – stream identifier in backend server database
- createDate – stream creation date
- endDate – stream termination date
- duration – длительность потока
- subscribers – list of identifiers of subscribers (forpublished streams)

If node Id is set, the response will contain only that node stream history information. If Id is not set, the response will contain all the streams history on backend server corresponding to the other query criteria.

If maximum stream count in DB selection is set to 0, all stream data will be selected from DB. This query can be terminated by timeout if streams count in DB is too big, so it is not recommended to send the query without limits.