

Obtaining stream information from backend server

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/s tream/list content- length:207 { "requestId ": "f2378c35 -e7ca-4241- b382- 29f696f0ddd b", "realm": "/ api/stream/ list", "payload": { "node": "", "mediaId": " f3ad4670- 8e83-11e9- 8386- dbc3d191a79 a", "name": "", "publisher"</pre>	<pre>MESSAGE destination :/user/serv ice content- type:applic ation/json; charset=UTF -8 subscription:sub-1 message- id:3-35 content- length:210 { "requestId ": "f2378c35 -e7ca-4241- b382- 29f696f0ddd b", "status":2 00, "reason": " SUCCESS", "payload": [{ "name": "tes</pre>	200 OK 400 Object not found 500 Persist exception

API

```
: "",  
"status": ""  
,  
"showClosed"  
: false,  
"nodeIp": "1  
92.168.1.5"  
,  
"nodeHostNa  
me": ""  
}  
}
```

```
t",  
"publisher"  
: null,  
"mediaId": "  
f3ad4670-  
8e83-11e9-  
8386-  
dbc3d191a79  
a",  
"age": 229,  
"status": "P  
UBLISHING"  
}  
]  
}
```

Response status

API	Request	Response	Response status
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- **status** – stream status (**PENDING**, **PUBLISHING**, **PLAYING**, **UNPUBLISHED**, **STOPPED**, **FAILED**)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	<pre>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 } }</pre>	<pre>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": 37694968573 }] }</pre>	200 OK 400 Object not found 500 Persist exception

API**Request****Response status**

```
56,  
  
"AUDIO_CODE  
C": 111,  
  
"time":  
"2019-06-  
14T10:27:37  
.644Z",  
  
"AUDIO_RATE  
": 29816,  
  
"seq": 14,  
  
"AUDIO_LOST  
": 41  
    },  
    {  
  
"AUDIO_SYNC  
":  
37694968572  
36,  
  
"AUDIO_CODE  
C": 111,  
  
"time":  
"2019-06-  
14T10:27:37  
.52Z",  
  
"AUDIO_RATE  
": 31048,  
  
"seq": 14,  
  
"AUDIO_LOST  
": 41  
    },  
    {  
  
"VIDEO_SYNC  
":  
37694968572  
21,  
  
"VIDEO_CODE  
C": 119,  
  
"VIDEO_NACK  
": 38,
```

API**Request****Response status**

```
"VIDEO_PLI"  
: 0,
```

```
"VIDEO_RATE"  
: 404488,
```

```
"VIDEO_WIDT  
H": 320,
```

```
"time":  
"2019-06-  
14T10:27:37  
.494Z",
```

```
"VIDEO_FPS"  
: 23,
```

```
"VIDEO_HEIG  
HT": 240,
```

```
"seq": 16  
    },  
    {
```

```
"AUDIO_SYNC"  
":  
37694968571  
16,
```

```
"AUDIO_CODE  
C": 111,
```

```
"time":  
"2019-06-  
14T10:27:37  
.397Z",
```

```
"AUDIO_RATE"  
: 29304,
```

```
"seq": 14,
```

```
"AUDIO_LOST"  
: 41  
    },  
    {
```

```
"VIDEO_SYNC"  
":  
37694968570  
29,
```

API

Request

Response status

```
"VIDEO_CODE  
C": 119,
```

```
"VIDEO_NACK  
": 38,
```

```
"VIDEO_PLI"  
: 0,
```

```
"VIDEO_RATE  
": 409056,
```

```
"VIDEO_WIDT  
H": 320,
```

```
"time":  
"2019-06-  
14T10:27:37  
.301Z",
```

```
"VIDEO_FPS"  
: 22,
```

```
"VIDEO_HEIG  
HT": 240,
```

```
"seq": 16  
},  
{
```

```
"AUDIO_SYNC  
":  
37694968569  
96,
```

```
"AUDIO_CODE  
C": 111,
```

```
"time":  
"2019-06-  
14T10:27:37  
.285Z",
```

```
"AUDIO_RATE  
": 28112,
```

```
"seq": 14,
```

```
"AUDIO_LOST  
": 41  
},  
{
```

API

Request

Response status

```
"AUDIO_SYNC" : 3769496856876,
"AUDIO_CODE_C" : 111,
"time" : "2019-06-14T10:27:37.16Z",
"AUDIO_RATE" : 28248,
"seq" : 14,
"AUDIO_LOST" : 41
},
{
"VIDEO_SYNC" : 3769496856806,
"VIDEO_CODE_C" : 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,
```


API**Request****Response status**

```
"seq": 16
  },
  {
    "AUDIO_SYNC
    ":
    37694968567
    56,
    "AUDIO_CODE
    C": 111,
    "time":
    "2019-06-
    14T10:27:37
    .039Z",
    "AUDIO_RATE
    ": 27424,
    "seq": 14,
    "AUDIO_LOST
    ": 41
    },
    {
      "AUDIO_SYNC
      ":
      37694968566
      36,
      "AUDIO_CODE
      C": 111,
      "time":
      "2019-06-
      14T10:27:36
      .923Z",
      "AUDIO_RATE
      ": 26888,
      "seq": 14,
      "AUDIO_LOST
      ": 41
      }
    ]
  }
```

API	Request	Response	Response Status
POST API	<pre> POST: /api/stream /metrics "application/json; charset=utf-8" { "node": "3" , "mediaId": "5d03b340-8e8c-11e9-92b7-23d635edebe6", "limit": 10 , "offset": 0 } </pre>	<pre> { "status": 200, "reason": "SUCCESS", "payload": [{ "AUDIO_SYNC": 3769496857356, "AUDIO_CODE C": 111, "time": "2019-06-14T10:27:37.644Z", "AUDIO_RATE": 29816, "seq": 14, "AUDIO_LOST": 41 }, { "AUDIO_SYNC": 3769496857236, "AUDIO_CODE C": 111, "time": "2019-06-14T10:27:37.52Z", "AUDIO_RATE": 31048, "seq": 14, </pre>	<p>Response object not found 500 Persist exception</p>

API	Request		Response status
		<pre>"AUDIO_LOST" : 41, }, { "VIDEO_SYNC" : 37694968572 21, "VIDEO_CODE" : 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" : 37694968571 16, "AUDIO_CODE" : 111, "time" : "2019-06-14T10:27:37.397Z",</pre>	

API	Request		Response status
		<pre>"AUDIO_RATE" : 29304, "seq" : 14, "AUDIO_LOST" : 41 }, { "VIDEO_SYNC" : 37694968570 29, "VIDEO_CODE" : 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" : 37694968569 96, "AUDIO_CODE" : 111,</pre>	

API	Request		Response status
		<pre>"time": "2019-06- 14T10:27:37 .285Z", "AUDIO_RATE ": 28112, "seq": 14, "AUDIO_LOST ": 41 }, { "AUDIO_SYNC ": 37694968568 76, "AUDIO_CODE C": 111, "time": "2019-06- 14T10:27:37 .16Z", "AUDIO_RATE ": 28248, "seq": 14, "AUDIO_LOST ": 41 }, { "VIDEO_SYNC ": 37694968568 06, "VIDEO_CODE C": 119, "VIDEO_NACK ": 37, "VIDEO_PLI" : 0,</pre>	

API	Request		Response status
		<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_CODE" C": 111, "time": "2019-06-14T10:27:37.039Z", "AUDIO_RATE" : 27424, "seq": 14, "AUDIO_LOST" : 41 }, { "AUDIO_SYNC" : 3769496856636, "AUDIO_CODE" C": 111, "time":</pre>	

API	Request	Response	Response status
		<pre> "2019-06- 14T10:27:36 .923Z", "AUDIO_RATE ": 26888, "seq": 14, "AUDIO_LOST ": 41 }] } </pre>	

Where:

- `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/s tream/histo ry content- length:191 { "requestId ": "4a700886 </pre>	<pre> 8MESSAGE destination :/user/serv ice content- type:applic ation/json; charset=UTF -8 subscription:sub-1 </pre>	200 OK 400 Object not found 500 Persist exception

API

```
-a559-42b7-  
b43d-  
82679d8573d  
6",  
  "realm":"/  
api/stream/  
history",  
  "payload":  
  {  
  
    "node":"3",  
  
    "mediaId":  
    "f3ad4670-  
8e83-11e9-  
8386-  
dbc3d191a79  
a",  
  
    "name": "",  
  
    "publisher"  
    : "",  
  
    "status": ""  
  
    "limit": 1,  
  
    "offset": 0  
  }  
}
```

Response status

```
message-  
id:3-39  
content-  
length:310  
  
{  
  "requestId"  
  : "4a700886  
-a559-42b7-  
b43d-  
82679d8573d  
6",  
  "status": 2  
00,  
  "reason": "  
SUCCESS",  
  "payload":  
  [  
    {  
  
      "id": 265,  
  
      "name": "tes  
t",  
  
      "status": "U  
NPUBLISHED"  
    },  
  
    {  
  
      "mediaId": "  
f3ad4670-  
8e83-11e9-  
8386-  
dbc3d191a79  
a",  
  
      "createDate"  
      : "2019-06-  
14T09:08:20  
.560+0000",  
  
      "endDate": n  
ull,  
  
      "publisher"  
      : 0,  
  
      "duration":  
      598658,  
  
      "subscriber  
s": [],  
      "node": 3
```


API

Request

```
}  
]  
}
```

Response status

API	Request	Response	Response status
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 (for published 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.