

Obtaining realtime stream information

- [STOMP client connection over Websocket to backend server](#)
- [Connection setup](#)
- [Metric acquisition rate and messages receiving frequency](#)

Realtime stream information can be obtained using [Websocket API](#).

STOMP client connection over Websocket to backend server

To connect to backend server for obtaining realtime metric values, do the following:

1. Establish [Websocket API](#) connection
2. Subscribe to /stream/nodeId/mediaId event by sending the message

```
SUBSCRIBE
id:sub-2
destination:/stream/3/ca77c700-8e86-11e9-8386-dbc3d191a79a

^@
```

Where

- 3 - node identifier in backend database
- ca77c700-8e86-11e9-8386-dbc3d191a79a- stream media session identifier

After that, client starts receiving STOMP messages in plain text

```
MESSAGE
destination:/stream/3/ca77c700-8e86-11e9-8386-dbc3d191a79a
content-type:application/json;charset=UTF-8
subscription:sub-2
message-id:3-50
content-length:1339

[{"VIDEO_SYNC":1560504547907,"VIDEO_CODEC":119,"VIDEO_NACK":0,"VIDEO_PLI":0,"VIDEO_RATE":571616,"VIDEO_WIDTH":320,"VIDEO_HEIGHT":240,"VIDEO_FPS":30,"timestamp":1560504547917},{ "AUDIO_SYNC":1560504547890,"AUDIO_CODEC":111,"AUDIO_RATE":26664,"timestamp":1560504547923,"AUDIO_LOST":0},{ "AUDIO_SYNC":1560504548010,"AUDIO_CODEC":111,"AUDIO_RATE":27040,"timestamp":1560504548040,"AUDIO_LOST":0},{ "VIDEO_SYNC":1560504548099,"VIDEO_CODEC":119,"VIDEO_NACK":0,"VIDEO_PLI":0,"VIDEO_RATE":577416,"VIDEO_WIDTH":320,"VIDEO_HEIGHT":240,"VIDEO_FPS":30,"timestamp":1560504548119},{ "AUDIO_SYNC":1560504548130,"AUDIO_CODEC":111,"AUDIO_RATE":26624,"timestamp":1560504548167,"AUDIO_LOST":0},{ "AUDIO_SYNC":1560504548250,"AUDIO_CODEC":111,"AUDIO_RATE":27976,"timestamp":1560504548282,"AUDIO_LOST":0},{ "VIDEO_SYNC":1560504548307,"VIDEO_CODEC":119,"VIDEO_NACK":0,"VIDEO_PLI":0,"VIDEO_RATE":581968,"VIDEO_WIDTH":320,"VIDEO_HEIGHT":240,"VIDEO_FPS":31,"timestamp":1560504548318},{ "AUDIO_SYNC":1560504548370,"AUDIO_CODEC":111,"AUDIO_RATE":28128,"timestamp":1560504548404,"AUDIO_LOST":0},{ "VIDEO_SYNC":1560504548499,"VIDEO_CODEC":119,"VIDEO_NACK":0,"VIDEO_PLI":0,"VIDEO_RATE":583032,"VIDEO_WIDTH":320,"VIDEO_HEIGHT":240,"VIDEO_FPS":31,"timestamp":1560504548506},{ "AUDIO_SYNC":1560504548490,"AUDIO_CODEC":111,"AUDIO_RATE":29568,"timestamp":1560504548522,"AUDIO_LOST":0}]
```

or in JSON using [STOMP.js](#) library

```
{
  "command": "MESSAGE",
  "headers": {
    "content-length": "1339",
    "message-id": "3-50",
    "subscription": "sub-2",
    "content-type": "application/json;charset=UTF-8",
    "destination": "/stream/3/ca77c700-8e86-11e9-8386-dbc3d191a79a"
  },
  "body": [
    {
      "VIDEO_SYNC": 1560504547907,
      "VIDEO_CODEC": 119,
      "VIDEO_NACK": 0,
      "VIDEO_PLI": 0,
```

```
"VIDEO_RATE": 571616,  
"VIDEO_WIDTH": 320,  
"VIDEO_HEIGHT": 240,  
"VIDEO_FPS": 30,  
"timestamp": 1560504547917  
},  
{  
  "AUDIO_SYNC": 1560504547890,  
  "AUDIO_CODEC": 111,  
  "AUDIO_RATE": 26664,  
  "timestamp": 1560504547923,  
  "AUDIO_LOST": 0  
},  
{  
  "AUDIO_SYNC": 1560504548010,  
  "AUDIO_CODEC": 111,  
  "AUDIO_RATE": 27040,  
  "timestamp": 1560504548040,  
  "AUDIO_LOST": 0  
},  
{  
  "VIDEO_SYNC": 1560504548099,  
  "VIDEO_CODEC": 119,  
  "VIDEO_NACK": 0,  
  "VIDEO_PLI": 0,  
  "VIDEO_RATE": 577416,  
  "VIDEO_WIDTH": 320,  
  "VIDEO_HEIGHT": 240,  
  "VIDEO_FPS": 30,  
  "timestamp": 1560504548119  
},  
{  
  "AUDIO_SYNC": 1560504548130,  
  "AUDIO_CODEC": 111,  
  "AUDIO_RATE": 26624,  
  "timestamp": 1560504548167,  
  "AUDIO_LOST": 0  
},  
{  
  "AUDIO_SYNC": 1560504548250,  
  "AUDIO_CODEC": 111,  
  "AUDIO_RATE": 27976,  
  "timestamp": 1560504548282,  
  "AUDIO_LOST": 0  
},  
{  
  "VIDEO_SYNC": 1560504548307,  
  "VIDEO_CODEC": 119,  
  "VIDEO_NACK": 0,  
  "VIDEO_PLI": 0,  
  "VIDEO_RATE": 581968,  
  "VIDEO_WIDTH": 320,  
  "VIDEO_HEIGHT": 240,  
  "VIDEO_FPS": 31,  
  "timestamp": 1560504548318  
},  
{  
  "AUDIO_SYNC": 1560504548370,  
  "AUDIO_CODEC": 111,  
  "AUDIO_RATE": 28128,  
  "timestamp": 1560504548404,  
  "AUDIO_LOST": 0  
},  
{  
  "VIDEO_SYNC": 1560504548499,  
  "VIDEO_CODEC": 119,  
  "VIDEO_NACK": 0,  
  "VIDEO_PLI": 0,  
  "VIDEO_RATE": 583032,  
  "VIDEO_WIDTH": 320,  
  "VIDEO_HEIGHT": 240,
```

```

        "VIDEO_FPS": 31,
        "timestamp": 1560504548506
    },
    {
        "AUDIO_SYNC": 1560504548490,
        "AUDIO_CODEC": 111,
        "AUDIO_RATE": 29568,
        "timestamp": 1560504548522,
        "AUDIO_LOST": 0
    }
]
}

```

The messages contain metric values acquired from the node where the stream is published or played.

To stop receiving metrics in realtime, the following message should be sent to unsubscribe

```

UNSUBSCRIBE
id:sub-2

^@

```

Connection setup

Maximum metrics count in one STOMP message is set with the following parameter in wcoam.properties

```
stomp_max_metrics=10
```

Connection timeout is set with the following parameter

```
stomp_max_timeout=1000
```

Metric aquisition rate and messages receiving frequency

A different metrics sets can be acquired to backend server [with a different rate](#) defined [by profile](#). If metric value, VIDEO_HEIGHT for example, does not change while stream is published, then messages with this metric will be send to subscriber at least as rate defines. If metric value (VIDEO_RATE) changes, messages with this metric can be send to subscriber upon changes.

For example, if stream data are collected by profile including one static metric VIDEO_WIDTH with rate 30

```

https://hostname:8090/api/profile/create
{
  "name": "profile1",
  "rate": "30",
  "metrics": ["2"],
  "rules": ["1"]
}

```

and maximum metrics count 1 per one message

```
stomp_max_metrics=1
```

the messages will be send to subscriber near 1 per second.

When changing rate to 600

```
https://hostname:8090/api/profile/update
{
  "id": "2",
  "name": "profile1",
  "rate": "600",
  "metrics": ["2"],
  "rules": ["1"]
}
```

the messages will be send to subscriber near 1 per 20 seconds.

Now, extending metrics set to 4 static metricsVIDEO_WIDTH, VIDEO_HEIGHT, VIDEO_CODEC, AUDIO_CODEC

```
https://hostname:8090/api/profile/update
{
  "id": "2",
  "name": "profile1",
  "rate": "600",
  "metrics": ["2","3","9","13"],
  "rules": ["1"]
}
```

the messages will be send to subscriber near 1 per 5 seconds.

When stream data are collected by profile including two metricsVIDEO_RATE, AUDIO_RATE with rate 30

```
https://hostname:8090/api/profile/create
{
  "name": "profile2",
  "rate": "30",
  "metrics": ["4","11"],
  "rules": ["1"]
}
```

and maximum metrics count 1 per one message

```
stomp_max_metrics=1
```

for the stream published to WCS using RTMP, the messages will be send to subscriber near 1-2 per second.