

# Testing with web interface

- [Profile addition](#)
- [Rule addition to a profile](#)
- [Node addition](#)
- [Obtaining current information about streams](#)

For backend server testing purposes, there is a web interface available by URL <https://hostname:8090/index.html>

## Profile addition

To add a profile, fill the Name, Rate, Rules and Metrics fields. Press "Profile new".

Profile list	Id: <input type="text"/>					
Profile new	Name: <input type="text" value="testprofile"/>	Rate: <input type="text" value="5"/>	Rules: <input type="text" value="13"/>	Metrics: <input type="text" value="4, 7"/>		
Profile update	Id: <input type="text"/>	Load by id	Name: <input type="text"/>	Rate: <input type="text"/>	Rules: <input type="text"/>	Metrics: <input type="text"/>
Profile delete	Id: <input type="text"/>					

Press "Profile list". A list of profiles will appear at the bottom of the page.

### Request:

```
POST: /api/profile/list "application/json; charset=utf-8"
{"id":""}
```

### Alarms:

### Response:

```
[
  {
    "id": 14,
    "name": "all",
    "rate": 5,
    "metrics": [
      1,
      2,
      3,
      4,
      5,
      6,
      7,
      8,
      9,
      10,
      11,
      12
    ],
    "rules": [
      13
    ]
  },
  {
    "id": 16,
    "name": "testprofile",
    "rate": 5,
    "metrics": [
      4,
      7
    ],
    "rules": [
      13
    ]
  }
]
```

## Rule addition to a profile

To add a rule, fill Name, Value, Profiles fields. Press "Rule new".

Rule list	Id: <input type="text"/>		ProfileId: <input type="text"/>	
Rule new	Type: <input type="text" value="1"/>	Name: <input type="text" value="testrule"/>	Value: <input type="text" value=".*"/>	Profiles: <input type="text" value="16"/>
Rule update	Id: <input type="text"/>		<input type="button" value="Load by id"/>	Type: <input type="text" value="Stream name"/> Name: <input type="text"/> Value: <input type="text"/> Profiles: <input type="text"/>
Rule delete	Id: <input type="text"/>			

Press "Rule list". A list of rules will appear at the bottom of the page.

### Request:

```
POST: /api/rule/list "application/json; charset=utf-8"
{"id":"","profileId":""}
```

### Alarms:

### Response:

```
[
  {
    "id": 13,
    "profiles": [
      14,
      16
    ],
    "type": 0,
    "name": "all",
    "value": ".*"
  },
  {
    "id": 17,
    "profiles": [
      16
    ],
    "type": 0,
    "name": "testrule",
    "value": ".*"
  }
]
```

## Node addition

To add a node, fill Ip, Port, ProfileList fields. Press "Node new".

Node list	Id: <input type="text"/>			
Node new	Ip: <input type="text" value="127.0.0.1"/>	Port: <input type="text" value="7777"/>	ProfileList: <input type="text" value="14"/>	
Node update	Id: <input type="text"/>	Load by id	Ip: <input type="text"/>	Port: <input type="text"/>
Node delete	Id: <input type="text"/>			
			Profiles:	<input type="text"/>

Press "Node list". A list of nodes will appear at the bottom of the page.

#### Request:

```
POST: /api/node/list "application/json; charset=utf-8"
{"id":""}
```

#### Alarms:

#### Response:

```
[
  {
    "id": 15,
    "host": "127.0.0.1",
    "port": 7777,
    "profiles": [
      14
    ]
  }
]
```

## Obtaining current information about streams

Open Two-way Streaming example, publish a stream. Set the node identifier and press "Stream cache list"

<div>Stream cache list</div>	Id: <input type="text" value="3"/>	MediaId: <input type="text"/>
----------------------------------	------------------------------------	-------------------------------

A list of streams will appear at the bottom of the page.

## Request:

```
POST: /api/stream/list "application/json; charset=utf-8"
{"node": "3", "mediaId": "", "name": "", "publisher": "", "status": ""}
```

## Alarms:

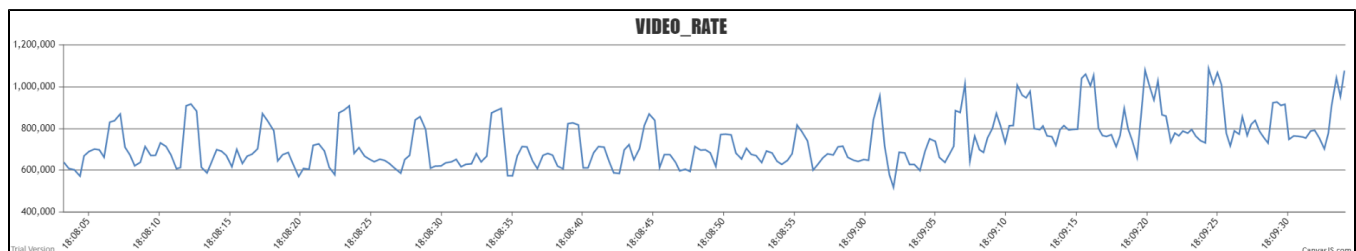
## Response:

```
[
  {
    "VIDEO_SYNC": 3767944032329,
    "AUDIO_SYNC": 3767944032334,
    "VIDEO_NACK": 17,
    "mediaId": "dee968a0-806d-11e9-90b4-df338db69136",
    "METRIC_RESOURCE": 0,
    "AUDIO_RATE": 31456,
    "STREAM_RESOURCE": 0,
    "AUDIO_LOST": 19,
    "VIDEO_CODEC": 119,
    "HEADER_RESOURCE": 0,
    "VIDEO_PLI": 0,
    "name": "test2",
    "publisher": null,
    "VIDEO_RATE": 843024,
    "AUDIO_CODEC": 111,
    "VIDEO_WIDTH": 320,
    "VIDEO_HEIGHT": 240,
    "VIDEO_FPS": 17,
    "age": 412,
    "KEEP_ALIVE": 0,
    "status": "PUBLISHING"
  }
]
```

Copy node и mediaId parameters to MediaId и NodeId fields, press "STOMP subscribe".

STOMP subscribe	STOMP unsubscribe	MediaId: <input type="text" value="dee968a0-806d-11e9-90b4-c"/>	NodeId: <input type="text" value="3"/>
Node status	Id: <input type="text"/>		

In a separate browser tab, video bitrate graph begins to bedrawn.



Press Close. The graph tab will be closed, STOMP messages with stream data will be stopped

Streams

NodeId:

MediaId:

Metrics

Id:

STOMP  
subscribe

MediaId:

123bdd40-85ec-11e9-aac3-8

NodeId:

3

Stomp Id	Media Id	Server Id	Action
sub-4	123bdd40-85ec-11e9-aac3-85bc07f60bba	3	<input type="button" value="Close"/>