Управление тревогами

Существует возможность в реальном времени получать изменения значений определенных метрик, если значение превысило или опустилось ниже заданного порога. Для этого создаются тревоги, для получения которых по протоколу STOMP через Websocket необходимо подписаться на событие `/alarms`

Управление тревогами осуществляется при помощи Websocket API или REST API.

Создание тревоги

Новая тревога создается при помощи запроса `/api/alarm/create`:

API	Request	Response	Response status

API	Request	Response	Response status

WS API	SEND
	destination:/app/api/alarm/create
	content-length:174
	
	{
	"requestId":"c71ec29d-b292-46c0-9138-a8ff434f2c3e",
	"realm":"/api/alarm/create",
	"payload":
	{
	"type":"0",
	"name":"alarm1",
	"value":"100000",
	"metric":"3",
	"node":"3",
	"time":"1000"
	}
	}
		MESSAGE
		destination:/user/service
		content-type:application/json;charset=UTF-8
		subscription:sub-1
		message-id:3-51
		content-length:84
		
		{
		"requestId":"c71ec29d-b292-46c0-9138-a8ff434f2c3e",
		"status":200,
		"reason":"SUCCESS"
		}
			200 OK 400 Object not found 500 Persist exception

API	Request	Response	Response status
REST API	POST: /api/alarm/create "application/json; charset=utf-8" { "type":"0", "name":"alarm2", "value":"100000", "metric":"3", "node":"3", "time":"1000" }	{ "status":200, "reason":"SUCCESS" }	200 OK 400 Object not found 500 Persist exception

Здесь:

- `type` – тип тревоги:
  - `0` – значение опустилось ниже заданного порога
  - `1` – значение превысило заданный порог
  - `2` - значение равно заданной величине
  - `4` - значение, монотонно возрастающее, опустилось
  - `5` - значение, монотонно снижающееся, возросло
- `name` – имя тревоги
- `value` – пороговое значение
- `metric` – идентификатор метрики (в данном примере битрейт видео)
- `node` – идентификатор узла
- `time` – время в миллисекундах, в течение которого условие срабатывания тревоги выполняется.

В данном примере создана тревога, срабатывающая, если битрейт видеоопубликованного на сервере потока опустится ниже 100 кбит/с более чем на 1 секунду.

Если идентификатор узла не указан, тревога применяется ко всем узлам на бэкенд-сервере.

На одну метрику может быть назначено несколько тревог, например, ограничивая нижний и верхний пределы битрейта видео.

Изменение тревоги

Параметры тревоги могут быть изменены при помощи запроса `/api/alarm/update`:

API	Request	Response	Response status

API	Request	Response	Response status

WS API	SEND
	destination:/app/api/alarm/update
	content-length:183
	
	{
	"requestId":"a60920eb-257a-451f-937f-12226a3856610",
	"realm":"/api/alarm/update",
	"payload":
	{
	"id":"6",
	
	"type":"0",
	
	"name":"alarm1",
	
	"value":"100000",
	
	"metric":"3",
	
	"node":"3",
	
	"time":"1000"
	}
	}
		MESSAGE
		destination:/user/service
		content-type:application/json;charset=UTF-8
		subscription:sub-1
		message-id:3-56
		content-length:84
		
		{
		"requestId":"a60920eb-257a-451f-937f-12226a3856610",
		"status":200,
		"reason":"SUCCESS"
		}
			200 OK 400 Object not found 500 Persist exception

API	Request	Response	Response status
REST API	POST: /api/alarm/update "application/json; charset=utf-8" { "id":"7", "type":"0", "name":"alarm2", "value":"10000", "metric":"3", "node":"3", "time":"1000" }	{ "status":200, "reason":"SUCCESS" }	200 OK 400 Object not found 500 Persist exception

Здесь:

- `id` – идентификатор тревоги
- `type` – тип тревоги:
  - `0` – значение опустилось ниже заданного порога
  - `1` – значение превысило заданный порог
  - `2` - значение равно заданной величине
  - `4` - значение, монотонно возрастающее, опустилось
  - `5` - значение, монотонно снижающееся, возросло
- `name` – имя тревоги
- `value` – пороговое значение
- `metric` – идентификатор метрики (в данном примере битрейт видео)
- `node` – идентификатор узла
- `time` – время в миллисекундах, в течение которого условие срабатывания тревоги выполняется.

Удаление тревоги

Тревога может быть удалена при помощи запроса `/api/alarm/delete`:

API	Request	Response	Response status

WS API	SEND
	destination:/app/api/alarm/delete
	content-length:101
	
	{
	"requestId":"c108dbf9-35c0-42e5-814c-0eec57c4de8e",
	"realm":"/api/alarm/delete",
	"payload":
	{
	"id":"6"
	}
	}
		MESSAGE
		destination:/user/service
		content-type:application/json;charset=UTF-8
		subscription:sub-1
		message-id:3-57
		content-length:84
		
		{
		"requestId":"c108dbf9-35c0-42e5-814c-0eec57c4de8e",
		"status":200,
		"reason":"SUCCESS"
		}
			200 OK 400 Object not found 500 Persist exception

REST API	POST:
	/api/alarm/delete
	"application/json;charset=utf-8"
	{
	"id":"7"
	}
		{
		"status":200,
		"reason":"SUCCESS"
		}
			200 OK 400 Object not found 500 Persist exception

Здесь:

- `id` – идентификатор тревоги

Сообщения о выходе метрики за допустимый предел больше не будут приходить.

Получение информации о настройках тревоги

Информацию о настройках тревоги можно получить при помощи запроса `/api/alarm/list`:

API	Request	Response	Response status

API	Request	Response	Response status

WS API	SEND
	destination:/app/api/alarm/list
	content-length:98
	
	{
	"requestId":"d8e79851-85eb-4df1-bd3a-9f13090e8be5",
	"realm":"/api/alarm/list",
	"payload":
	{
	"id":""
	}
	}
		MESSAGE
		destination:/user/service
		content-type:application/json;charset=UTF-8
		subscription:sub-1
		message-id:3-60
		content-length:177
		
		{
		"requestId":"d8e79851-85eb-4df1-bd3a-9f13090e8be5",
		"status":200,
		"reason":"SUCCESS",
		"payload":
		[
		{
		"id":8,
		
		"name":"alarm1",
		
		"type":0,
		
		"value":100000,
		
		"time":1000,
		
		"metric":3,
		"node":3
		}
		]
		}
			200 OK 400 Object not found 500 Persist exception

API	Request	Response	Response status
REST API	POST: /api/alarm/list "application/json; charset=utf-8" { "id":"" }	{ "status": 200, "reason": "SUCCESS", "payload": [{ "id": 8, "name": "alarm1", "type": 0, "value": 100000, "time": 1000, "metric": 3, "node": 3 }] }	200 OK 400 Object not found 500 Persist exception

Здесь:

- `id` – идентификатор тревоги
- `type` – тип тревоги:
  - `0` – значение опустилось ниже заданного порога
  - `1` – значение превысило заданный порог
  - `2` - значение равно заданной величине
  - `4` - значение, монотонно возрастающее, опустилось
  - `5` - значение, монотонно снижающееся, возросло
- `name` – имя тревоги

- `value` – пороговое значение
- `metric` – идентификатор метрики (в данном примере битрейт видео)
- `node` – идентификатор узла
- `time` – время в миллисекундах, в течение которого условие срабатывания тревоги выполняется.

Если указан идентификатор тревоги, то ответ будет содержать информацию только об этой тревоге. Если идентификатор не указан, ответ будет содержать информацию обо всех тревогах на бэкенд-сервере.

Состав полей ответа аналогичен составу полей запроса `/api/alarm/update`.

Получение сообщения о срабатывании тревоги

Сообщения о срабатываниях и возвратах тревог приходят, если клиент подписан на очередь `/alarms`. Сообщения выглядят следующим образом:

```
MESSAGE
destination:/alarms
content-type:application/json;charset=UTF-8
subscription:sub-0
message-id:4-187
content-length:242

{
 "timestamp":1561101716609,
 "status":"RAISED",
 "alarmType":"LESS",
 "alarmValue":700000,
 "alarmName":"alarm1",
 "mediaId":"617691c0-93f2-11e9-8808-938c74814152",
 "metricEnumName":"VIDEO_RATE",
 "metricValue":400232,
 "nodeHostName":"test.flashphoner.com"
}
```

Здесь:

- `timestamp` - время срабатывания или возврата тревоги
- `status` - состояние тревоги:
  - `RAISED` - срабатывание
  - `CLEARED` - возврат
- `alarmType` - тип тревоги:
  - `LESS` – значение опустилось ниже заданного порога
  - `MORE` – значение превысило заданный порог

    - `EQUAL` - значение равно заданной величине
    - `MONOTONIC_UP` - значение, монотонно возрастающее, опустилось
    - `MONOTONIC_DOWN` - значение, монотонно снижающееся, возросло
- `alarmValue` - пороговое значение
- `alarmName` - имя тревоги
- `mediaId` - идентификатор медиасессии потока, для которого зафиксировано событие
- `metricEnumName` - наименование метрики
- `metricValue` - значение метрики, по которому сработала или вернулась тревога
- `nodeHostName` - имя узла, на котором находится наблюдаемый поток