

# Описание REST API

- [API адреса](#)
- [Настройки](#)
  - [Порты](#)
  - [Междоменные запросы](#)
  - [Авторизация](#)
- [Тестирование](#)
- [OpenAPI спецификация](#)

## API адреса

Web Call Server использует два URL-адреса для REST API

HTTP	HTTPS
<a href="http://host:8081/rest-api">http://host:8081/rest-api</a>	<a href="https://host:8444/rest-api">https://host:8444/rest-api</a>

Для того чтобы вызвать REST-метод, сформируйте запрос следующего формата:

- Протокол: HTTP или HTTPS
- Content Type: application/json
- Метод: POST
- Тело запроса: JSON

## Настройки

### Порты

По умолчанию, REST HTTP запросы принимаются на порту 8081, REST HTTPS - на порту 8444. Эти значения определяются следующими настройками в файле [flashphoner.properties](#)

```
http.port=8081
https.port=8444
```

### Междоменные запросы

Междоменные запросы управляются следующими настройками в файле [flashphoner.properties](#)

```
rest_access_control_allow_origin=*
rest_access_control_allow_headers=content-type,x-requested-with
rest_access_control_allow_methods=POST
```

По умолчанию разрешены междоменные запросы с любого адреса, чтобы обеспечить перенаправление портов.

### Авторизация

По умолчанию, REST API запросы обрабатываются без авторизации. При необходимости, может быть включена Basic авторизация при помощи следующего параметра в файле [flashphoner.properties](#)

```
disable_rest_auth=false
```

Пользователи REST API при включении авторизации управляются из [командной строки ядра WCS](#).

Для междоменных запросов необходимо добавить заголовки при помощи следующей настройки в файле [flashphoner.properties](#):

```
rest_access_control_headers=Access-Control-Allow-Origin: <origin>;Access-Control-Allow-Credentials: true;Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS;Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept, Authorization
```

Здесь <origin>- источник запроса, например <http://test.flashphoner.com>.

## Тестирование

Для тестирования можно использовать REST-клиент. Например [Advanced REST Console](#) для браузера Google Chrome.

На скриншоте приведен пример обращения к серверу с получением списка текущих потоков REST методом `stream/find_all`

Method: POST URL: [http://demo.flashphoner.com:8081/rest-api/stream/find\\_all](http://demo.flashphoner.com:8081/rest-api/stream/find_all) SEND

HEADERS BODY AUTHORIZATION VARIABLES

Name	Value
Content-Type	application/json

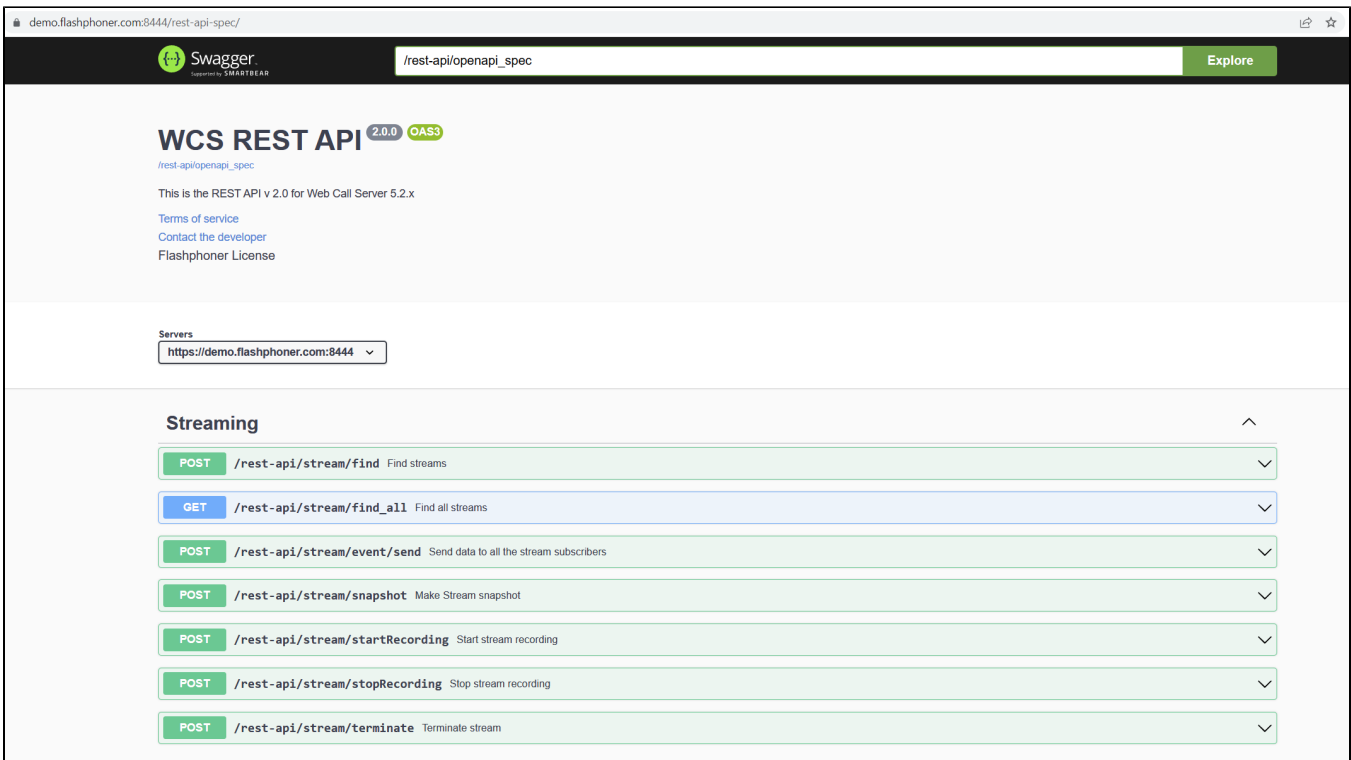
Response 200 OK 1008 B 355 ms

```
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 904
```

```
1 - [
2   {
3     "appKey": "flashStreamingApp",
4     "sessionId": "/36.71.175.77:63690/188.40.244.249:1935",
5     "mediaSessionId": "02d63bad-0a05-4ea6-b3be-c616cd44529d",
6     "name": "123",
7     "published": true,
8     "hasVideo": true,
9     "hasAudio": true,
10    "status": "PUBLISHING",
11    "sdp": "v=0\r\no=- 1988962254 1988962254 IN IP4 0.0.0.0\r\nnc=IN IP4 0.0.0.0\r\nnt=0 0\r\nna=sdplang:en\r\nnm=video 0 RTP/AVP 119
127\r\nna=rtmpmap:119 H264/90000\r\nna=fmt:119 profile-level-id=42e01f;packetization-mode=1\r\nna=rtmpmap:127 FLV/90000\r\nna
=sendonly\r\nna=audio 0 RTP/AVP 97 8 0\r\nna=rtmpmap:97 SPEEX/16000\r\nna=rtmpmap:8 PCMA/8000\r\nna=rtmpmap:0 PCMU/8000\r\nna
=sendonly\r\n",
12    "audioCodec": "mpeg4-generic",
13    "videoCodec": "H264",
14    "record": true,
15    "recordName": "stream-02d63bad-0a05-4ea6-b3be-c616cd44529d-b761as0rig0jyvboenn0d0jjo.mp4",
16    "width": 1920,
17    "height": 1080,
18    "bitrate": 0,
19    "minBitrate": 0,
20    "maxBitrate": 0,
21    "quality": 0,
22    "history": false,
```

## OpenAPI спецификация

В сборке [5.2.1343](#) добавлен интерфейс для отображения и тестирования REST API в соответствии с OpenAPI 3 спецификацией. Файл OpenAPI спецификации может быть загружен по адресу [https://demo.flashphoner.com:8444/rest-api/openapi\\_spec](https://demo.flashphoner.com:8444/rest-api/openapi_spec) в формате YAML, интерфейс доступен по адресу <https://demo.flashphoner.com:8444/rest-api-spec/>, где `demo.flashphoner.com` - адрес WCS сервера



Каждый запрос можно развернуть и протестировать при помощи Swagger UI



Обязательные поля отмечены символом \* на вкладке Schema

# Streaming



**POST** /rest-api/stream/find Find streams



Find streams by filter

### Parameters

Try it out

No parameters

Request body **required**

application/json



Example Value | Schema

```
StreamFilter {
  custom > {...}
  nodeId > {...}
  appKey string
  sessionId string
  mediaSessionId* string
  name* string
  published example: streamI boolean
  status string
  Enum:
  display > Array [ 11 ]
  customMap > [...]
  > {...}
}
```