

# API overview

## API URLs

Web Call Server uses two URLs for REST API

HTTP	HTTPS
<code>http://host:8081/rest-api</code>	<code>https://host:8444/rest-api</code>

To invoke the REST API, you have to make request using the following format:

- Protocol: HTTP or HTTPS
- Content Type: application/json
- Method: POST
- Body: JSON

## Settings

### Ports

By default, server accepts REST HTTP requests on 8081 TCP port and REST HTTPS requests on 8444 TCP port. Those values are defined with the following settings in [flashphoner.properties](#) file

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

### Cross-domain requests

Cross-domain request are managed with the following settings in [flashphoner.properties](#) file

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

By default, cross-domain requests from any address are allowed, to make port redirection work.

## Authorization

REST API requests are handled without user authorization by default. HTTP basic authorization may be enabled with the following parameter in [flashphoner.properties](#) file if necessary

```
disable_rest_auth=false
```

When authorization is enabled, REST API users are managed from [WCS core CLI](#).

For cross-domain requests, custom headers have to be added using the following parameter in [flashphoner.properties](#) file:

```
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
```

Here `origin` is the request source, e.g. `http://test.flashphoner.com`.

## Testing

For testing purpose you can use a REST client, for example [Advanced REST Console](#) for browser Google Chrome.

On the screenshot below you can see how to get list of streams by REST API using REST method `/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

```

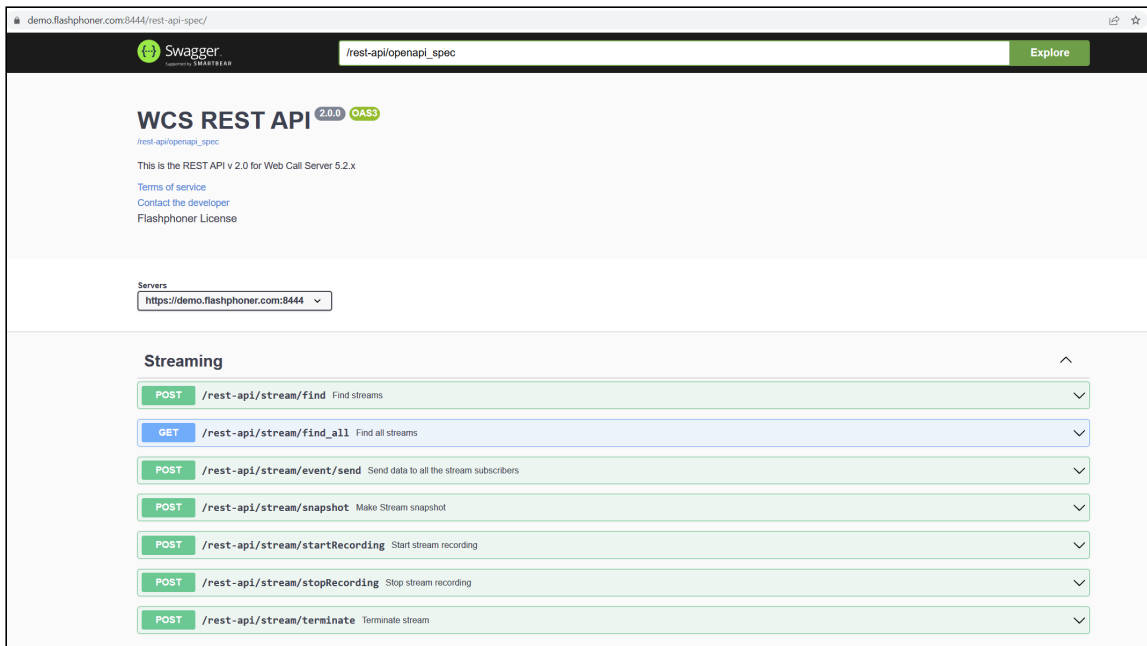
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 904
  
```

```

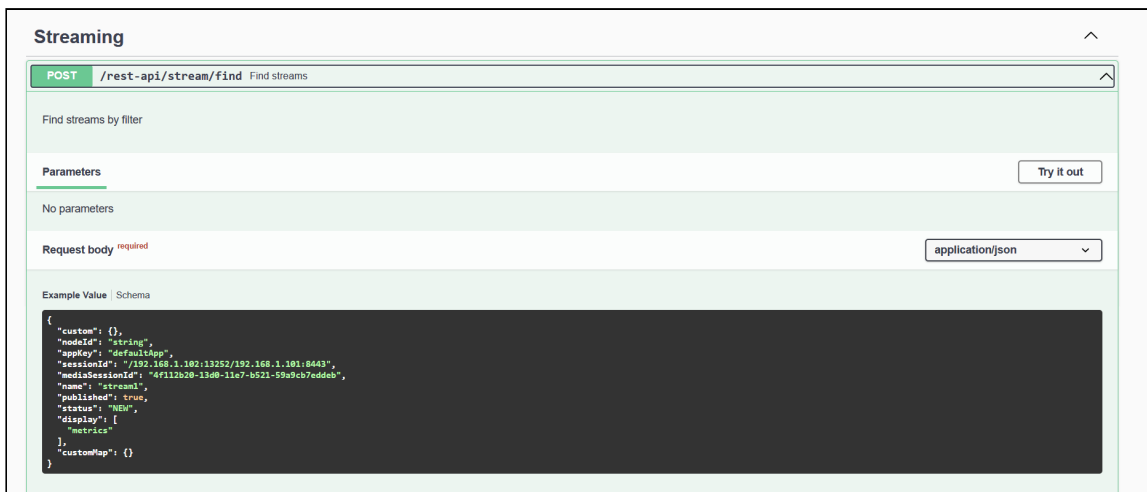
1  [
2  {
3    "appKey": "flashStreamingApp",
4    "sessionId": "/36.71.175.77:63690/188.40.244.249:1935",
5    "mediaSessionId": "02d63bad-0a05-4ea6-b3be-c616cd44529d",
6    "name": "123",
7    "published": true,
8    "hasVideo": true,
9    "hasAudio": true,
10   "status": "PUBLISHING",
11   "sdp": "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\na=sdplang:en\r\nm=video 0 RTP/AVP 119
127\r\na=rtmpmap:119 H264/90000\r\na=fmtp:119 profile-level-id=42e01f;packetization-mode=1\r\na=rtmpmap:127 FLV/90000\r\na
=sendonly\r\nm=audio 0 RTP/AVP 97 8 0\r\na=rtmpmap:97 SPEEX/16000\r\na=rtmpmap:8 PCMA/8000\r\na=rtmpmap:0 PCMU/8000\r\na
=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 specification

Since build 5.2.1343 REST API can be displayed and tested according to OpenAPI 3 specification. OpenAPI specification file can be downloaded by URL [https://demo.flashphoner.com:8444/rest-api/openapi\\_spec](https://demo.flashphoner.com:8444/rest-api/openapi_spec) in YAML format, UI is available by URL <https://demo.flashphoner.com:8444/rest-api-spec/>, where `demo.flashphoner.com` is WCS server address



Every REST API query can be explored and tested using Swagger UI



Required fields are marked by \* character on Schema tab.

## Streaming

POST /rest-api/stream/find Find streams

Find streams by filter

### Parameters

Try it out

No parameters

Request body <sup>required</sup>

application/json

Example Value | Schema

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