

API overview

- [API URLs](#)
- [Settings](#)
 - [Ports](#)
 - [Authorization](#)
 - [Cross-domain requests](#)
- [Testing](#)

API URLs

Web Call Server uses two URLs for REST API

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

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
```

In WebCallServer previous versions, the manager module handles REST queries on 9091 and 8888 ports respectively. For existing web applications to work after server update, the following parameters are provided in [wcs-manager.properties](#) file

```
-Dmanager.rest_redirect_http_port=8081
-Dmanager.rest_redirect_https_port=8444
```

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](#).

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.

Testing

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

On the screenshot below you can see how to get list of streams by REST method/stream/find_all

The screenshot displays the Advanced REST Client interface. At the top, the Method is set to POST and the URL is http://demo.flashphoner.com:8081/rest-api/stream/find_all. The Headers tab is active, showing Content-Type: application/json. The Response tab shows a 200 OK status with a response size of 1008 B and a response time of 355 ms. The response body is a JSON object containing stream details.

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

{
  "appKey": "flashStreamingApp",
  "sessionId": "/36.71.175.77:63690/188.40.244.249:1935",
  "mediaSessionId": "02d63bad-0a05-4ea6-b3be-c616cd44529d",
  "name": "123",
  "published": true,
  "hasVideo": true,
  "hasAudio": true,
  "status": "PUBLISHING",
  "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\nnm=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",
  "audioCodec": "mpeg4-generic",
  "videoCodec": "H264",
  "record": true,
  "recordName": "stream-02d63bad-0a05-4ea6-b3be-c616cd44529d-b761as0rig0jjvboeenn0d0jjo.mp4",
  "width": 1920,
  "height": 1080,
  "bitrate": 0,
  "minBitrate": 0,
  "maxBitrate": 0,
  "quality": 0,
  "history": false,
```