

# Описание REST API

## API адреса

Web Call Server использует два URL-адреса для REST API

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

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`

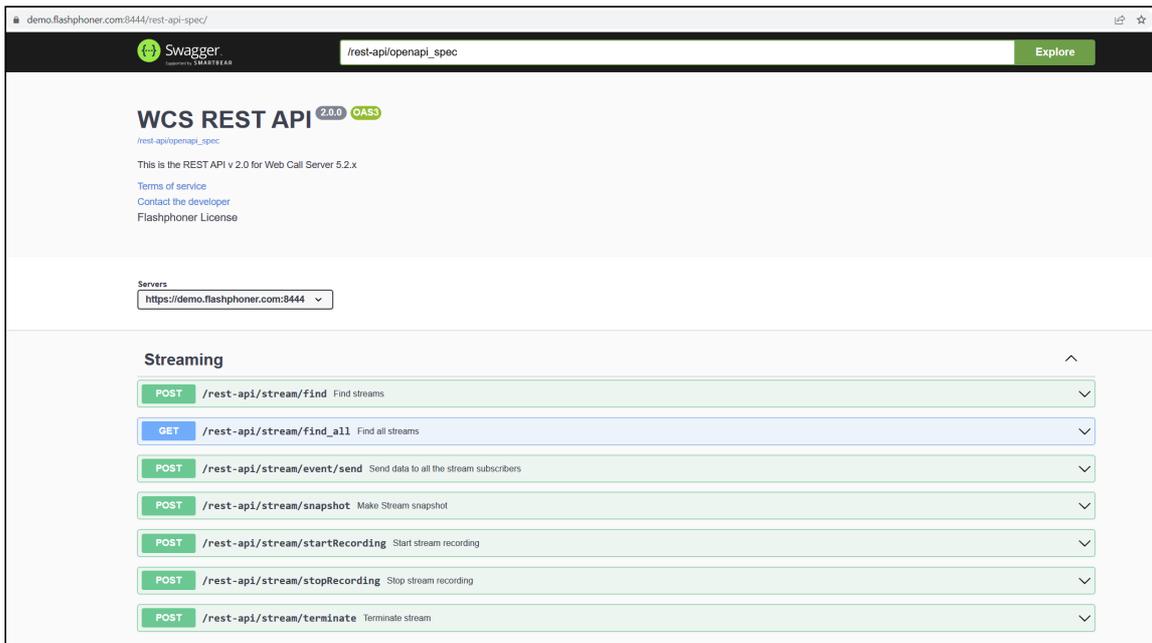
The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** [http://demo.flashphoner.com:8081/rest-api/stream/find\\_all](http://demo.flashphoner.com:8081/rest-api/stream/find_all)
- Headers:** Content-Type: application/json
- Response:** 200 OK, 1008 B, 355 ms
- Response Headers:** Access-Control-Allow-Origin: \*, Content-Type: application/json, Content-Length: 904
- Response Body (JSON):**

```
1  {}
2
3  {
4    "appKey": "flashStreamingApp",
5    "sessionId": "/36.71.175.77:63690/188.40.244.249:1935",
6    "mediaSessionId": "02d63bad-0a05-4ea6-b3be-c616cd44529d",
7    "name": "123",
8    "published": true,
9    "hasVideo": true,
10   "hasAudio": true,
11   "status": "PUBLISHING",
12   "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
13   127\r\na=rtpmap:119 H264/90000\r\na=fmtp:119 profile-level-id=42e01f;packetization-mode=1\r\na=rtpmap:127 FLV/90000\r\na
14   =sendonly\r\na=audio 0 RTP/AVP 97 8 0\r\na=rtpmap:97 SPEEX/16000\r\na=rtpmap:8 PCMA/8000\r\na=rtpmap:0 PCMU/8000\r\na
15   =sendonly\r\n",
16   "audioCodec": "mpeg4-generic",
17   "videoCodec": "H264",
18   "record": true,
19   "recordName": "stream-02d63bad-0a05-4ea6-b3be-c616cd44529d-b761as0rig0jyvboenn0d0jjo.mp4",
20   "width": 1920,
21   "height": 1080,
22   "bitrate": 0,
23   "minBitrate": 0,
24   "maxBitrate": 0,
25   "quality": 0,
26   "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 boolean
  status string
  Enum:
  display > Array [ 11 ]
  customMap > {...}
}
```