

Stream availability for playback

- [WebSDK function Stream.available\(\)](#)
- [REST API query /stream/find](#)

Sometimes it is necessary to check if stream is published on server before playing the stream. This can be done by one of the following ways

WebSDK function Stream.available()

Stream availability can be checked using WebSDK function `Stream.available()`. This function returns a promise that will be resolved if stream with the given name is published on server, and will be rejected if not. A special stream should be created to invoke the function, stream name to check and page element to display should be passes to `Session.createStream()` function. This special stream will automatically be destroyed after promise resolution or rejection,

`Stream.available()` usage example is available in [Two Way Streaming](#) application source code:

```
function availableStream(){
    var session = Flashphoner.getSessions()[0];
    var streamName = $('#playStream').val();
    session.createStream({
        name: streamName,
        display: remoteVideo
    }).available().then(function(stream){
        $('#availableStatus').text("AVAILABLE").attr("class", "text-success");
    }, function(stream){
        $('#availableStatus').text("UNAVAILABLE").attr("class", "text-danger");
    });
}
```

This method works in CDN, including [transcoded](#) streams, if stream name to check is passed with profile name. In this case promise can be rejected if stream is available on server, but is protected with [ACL key](#).

REST API query /stream/find

If the first method cannot be used for some reason, stream availability on server can be checked with [REST API](#) query `/stream/find`. The request should be sent directly to the server stream supposed to be played from. If the stream with given name is published on this server, it will respond 200 OK:

Request	Response
---------	----------

```
POST /rest-api/stream
/find HTTP/1.1
Host: 192.168.1.101:8081
Content-Length: 40
Content-Type: application
/json
```

```
{
    "name": "stream1",
    "published": true
}
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json;charset=UTF-8
Date: Tue, 28 Mar 2017 16:05:16 GMT
```

```
[
    {
        "custom": {},
        "nodeId": null,
        "appKey": "defaultApp",
        "sessionId": "/192.168.1.102:34500/192.168.1.101:8080",
        "mediaSessionId": "4f112b20-13d0-11e7-b521-59a9cb7eddeb",
        "remoteMediaElementId": null,
        "name": "stream1",
        "published": true,
        "hasVideo": true,
        "hasAudio": true,
        "status": "PUBLISHING",
        "sdp": "....",
        "info": null,
        "record": true,
        "recordName": "stream-4f112b20-13d0-11e7-b521-59a9cb7eddeb-7371rm8t053nlg8c2n23ctco33.mp4",
        "width": 0,
        "height": 0,
        "bitrate": 0,
        "quality": 0,
        "rtmpUrl": null,
        "streamInfo": {
            "custom": {},
            "nodeId": null,
            "appKey": null,
            "sessionId": null,
            "mediaSessionId": "4f112b20-13d0-11e7-b521-59a9cb7eddeb",
            "name": "stream1",
            "samplingTime": 1490717116551,
            "recordTimestamp": 7640,
            "recordStarted": true
        },
        "mediaProvider": "Flash"
    }
]
```

This method works only for streams already published on the requested server.