

# AgoraRTC Web SDK API Reference

The AgoraRTC JS SDK library includes the following classes.

<a href="#">AgoraRTC</a>	Use the AgoraRTC object to create Client and Stream objects
<a href="#">Client</a>	Represents the web client object which provides access to core AgoraRTC functionality.
<a href="#">Stream</a>	Represents the local/remote media stream in conversation.

## **AgoraRTC**

### Methods:

Name	Description
<code>createClient()</code>	create and returns a Client object
<code>createStream(spec)</code>	create and returns a Stream object

### `createClient()`

This method creates and returns a Client object, should be called only once.

```
var client = AgoraRTC.createClient();
```

### `createStream(spec)`

This method creates and returns a Stream object.

Parameters:

Name	Type	Description
spec	Object	<p>This object contains the following properties:</p> <p><u>streamID</u>: represents the stream ID, normally set to uid which can be retrieved from Client join callback.</p> <p><u>audio</u>: (flag) true/false, marks whether this stream contains audio track.</p> <p><u>video</u>: (flag) true/false, marks whether this stream contains video track.</p> <p><u>screen</u>: (flag) true/false, marks whether this stream contains screen sharing track, should be set to false in current version.</p> <p><u>attributes</u>: (optional) this object contains the following properties:</p> <ul style="list-style-type: none"> <li>- resolution: set to one of {'sif', 'vga', 'hd720p'}</li> <li>- minFrameRate: minimal video frame rate</li> <li>- maxFrameRate: maximum video frame rate</li> </ul>

```
var client = AgoraRTC.createClient({streamID: uid, audio:true, video:true, screen:false});
```

## Client

### Methods:

Name	Description
init(key, onSuccess, onFailure)	initialize the Client object.
join(token, channel, onSuccess, onFailure)	this method joins a AgoraRTC channel.
leave(onSuccess, onFailure)	this method leaves a AgoraRTC channel.
publish(stream, onFailure)	this method publishes local stream to server
unpublish(stream, onFailure)	this method unpublishes local stream
subscribe(stream, onFailure)	this method subscribes remote stream from server
unsubscribe(stream, onFailure)	this method unsubscribes remote stream

init(key, onSuccess, onFailure)

This method initializes the Client object.

Parameters:

Name	Type	Description
key	String	This is the vendor key provided by Agora.
onSuccess	function	(optional) the function to be called when the method succeeds.
onFailure	function	(optional) the function to be called when the method fails.

```

client.init(vendorKey, function() {
  log("client initialized");
  //join channel
  .....
}, function(err) {
  log("client init failed ", err);
  //error handling
});

```

---

join(token, channel, onSuccess, onFailure)

This method joins a AgoraRTC channel.

Parameters:

Name	Type	Description
token	String	This is the dynamic key which will take be used in later version, can be set to undefined in current version.
channel	String	A string represents the conversation channel.
onSuccess	function	(optional) the function to be called when the method succeeds, will return the uid which represents the identity of user.
onFailure	function	(optional) the function to be called when the method fails.

```

client.join(undefined, '1024abc', function(uid) {
  log("client " + uid + " joined channel");
  //create local stream
  .....
}, function(err) {
  log("client join failed ", err);
  //error handling
});

```

---

leave(onSuccess, onFailure)

This method leaves a AgoraRTC channel.

Parameters:

Name	Type	Description
onSuccess	function	(optional) the function to be called when the method succeeds.
onFailure	function	(optional) the function to be called when the method fails.

```
client.leave(function() {
  log("client leaves channel");
  .....
}, function(err) {
  log("client leave failed ", err);
  //error handling
});
```

---

publish(stream, onFailure)

This method publishes local stream to server.

Parameters:

Name	Type	Description
stream	object	Stream object which represents local stream.
onFailure	function	(optional) the function to be called when the method fails.

```
client.publish(stream, function(err) {
  log("stream published");
  .....
})
```

---

unpublish(stream, onFailure)

This method unpublishes local stream.

Parameters:

Name	Type	Description
stream	object	Stream object which represents local stream.
onFailure	function	(optional) the function to be called when the method fails.

```
client.unpublish(stream, function(err) {
  log("stream unpublished");
  .....
})
```

---

## subscribe(stream, onFailure)

This method subscribes remote stream from server.

Parameters:

Name	Type	Description
stream	object	Stream object which represents remote stream.
onFailure	function	(optional) the function to be called when the method fails.

```
client.subscribe(stream, function(err) {  
  log("stream unpublished");  
  .....  
})
```

---

## unsubscribe(stream, onFailure)

This method unsubscribes remote stream.

Parameters:

Name	Type	Description
stream	object	Stream object which represents remote stream.
onFailure	function	(optional) the function to be called when the method fails.

```
client.unsubscribe(stream, function(err) {  
  log("stream unpublished");  
  .....  
})
```

## Events:

---

### stream-published

Notify the local stream has been published.

```
client.on('stream-published', function(evt) {  
  log("local stream published");  
  .....  
})
```

## stream-added

Notify the remote stream has been added.

```
client.on('stream-added', function(evt) {  
  var stream = evt.stream;  
  log("new stream added ", stream.getId());  
  //subscribe the stream  
  .....  
})
```

---

## stream-subscribed

Notify the remote stream has been subscribed.

```
client.on('stream-subscribed', function(evt) {  
  var stream = evt.stream;  
  log("new stream subscribed ", stream.getId());  
  //play the stream  
  .....  
})
```

---

## ***Stream***

Methods:

Name	Description
init(onSuccess, onFailure)	initialize the Stream object.
getId()	this method retrieves stream id.
getAttributes()	this method retrieves stream attributes.
hasVideo()	this method retrieves the video flag.
hasAudio()	this method retrieves the audio flag.
enableVideo()	this method enables video track in the stream
disableVideo()	this method disables video track in the stream
enableAudio()	this method enables audio track in the stream
disableAudio()	this method disables audio track in the stream
setVideoResolution(width, height)	this method sets the video resolution.
setVideoFrameRate(framerate)	this method sets the video frame rate.
play(elementID)	this method plays the video/audio stream.

---

init(onSuccess, onFailure)

This method initializes the Stream object.

Parameters:

Name	Type	Description
onSuccess	function	(optional) the function to be called when the method succeeds.
onFailure	function	(optional) the function to be called when the method fails.

```
stream.init(function() {
    log("local stream initialized");
    // publish the stream
    .....
}, function(err) {
    log("local stream init failed ", err);
    //error handling
});
```

---

getId()

This method retrieves stream id.

---

### getAttributes()

This method retrieves stream attributes.

---

### hasVideo()

This method retrieves video flag.

---

### hasAudio()

This method retrieves audio flag.

---

### enableVideo()

This method enables video track in the stream, only works in the case when video flag was set to true in stream create, acts like video resume.

---

### disableVideo()

This method disables video track in the stream, only works in the case when video flag was set to true in stream create, acts like video pause.

---

### enableAudio()

This method enables audio track in the stream, acts like audio resume.

---

### disableAudio()

This method disables audio track in the stream, acts like audio pause.

---

### setVideoResolution(resolution)

This method sets the video resolution, it is optional, only works before calling stream.init().

Parameters:

Name	Type	Description
resolution	String	video resolution: can be set to one of the {'sif', 'vga', 'hd720p'}, represents 320x240, 640x480, 1280x720.

```
stream.setVideoResolution('vga');
```



---

## setVideoFrameRate(framerate)

This method sets the video frame rate, it is optional, only works before calling `stream.init()`.

Parameters:

Name	Type	Description
framerate	Object	set the min and max of video framerate.

```
stream.setVideoFrameRate([15, 30]);
```

---

## play(elementID)

This method plays the video/audio stream.

Parameters:

Name	Type	Description
elementID	String	represents the html element id.

```
stream.play('div_id');
```