Agora Whiteboard SDK Reference Manual v1.0

support@agora.io

Contents

Introduction

Agora CaaS

Agora Communications as a Service (CaaS) provides ensured Quality of Experience for worldwide, Internet-based voice and video communications through the Agora Global Network. The network optimizes real-time, mobile communications and solves quality of experience challenges for mobile devices in networks such as 3G/4G/Wi-Fi that perform erratically as well as Internet bottlenecks worldwide.

Agora CaaS includes the following SDKs and the applications using the Native SDK linked directly into the application when they are built:

- **Agora Native SDK for iOS** and **Agora Native SDK for Android**

  Mobile-optimized for smartphones, allowing access to the Agora Global Network along with device-specific mobile optimizations.
- **Agora Native SDK for Windows** and **Agora Native SDK for Mac**
- **Agora Native SDK for Web**

  The SDK provides web browsers with open access to the Agora Global Network.

Agora Whiteboard SDK

The Agora Whiteboard SDK is an addition to the Agora CaaS capabilities. The SDK provides a simple collaboration platform on the whiteboard where users from different locations can draw, annotate, and share PDF documents to visualize and simplify the communication. The SDK provides open access to the Agora Global Network from any device that supports a standard HTML5-compliant web browser, without requiring any downloads or plugins. It supports all plaforms.

The Agora Whiteboard SDK is a JavaScript library loaded by an HTML web page. It also provides a set of simple high-level JavaScript APIs for establishing whiteboard communications with users across the Agora Global Network.

The Agora Whiteboard SDK allows your JavaScript code to:

- **Establish Session** – Join and leave the shared Agora sessions (identified by unique channel names), where there may be many global users in a conference together.
- **Set user roles in the shared sessions** – host or attendee. Host can draw and annotate on the whiteboard, while attendees have view-only permission.

The application developers can access the Agora Whiteboard service in two ways:

- Web Application developers can use the JavaScript Class (Agora.Whiteboard) to join or leave a session. See Realizing Whiteboard

Function – Web Application and Agora Whiteboard SDK - API Referencefor details.

- Non-Web Application Developers(on any of the following platforms: iOS, Windows, Mac or Android) can use WebView to access the Agora Whiteboard URL service. See Realizing the Whiteboard Function – Non-Web Application for details.

Requirements

Compatibility

Agora Whiteboard SDK is compatible with all platform SDK versions.

Supported Web Browsers

The Agora Whiteboard SDK requires a HTML5-compliant web browser and has been tested and verified on all the following browser types and versions, which are available across a wide range of platforms.

	IE v10-11	**Chrome v49-51**	**Safari V9.1**	**Firefox v46-47**	**Opera v38**
HTTP/ HTTPS	Y	Y	Y	Y	Y

Note

All the browser versions listed have been tested and verified. Earlier versions of these browsers that support HTML5 may work, but have not been verified by Agora.io.

Supported File Formats

You can upload PDFs, office files and pictures in any of the following formats to the whiteboard:

File	**Format**
PDF	• .pdf
Excel	• .xls • .xlsx
Word	• .doc • .docx
PowerPoint	• .ppt • .pptx
Picture	• .png • .jpg • .jpeg

Note
When uploading pictures, the format JPEG2000 is not supported.

Required Documents

The Whiteboard SDK only provides the whiteboard functions for users to draw, upload files and annotate. If you want to realize the ordinary video or audio call functions, refer to each SDK reference manual.

All platform SDK and related reference manuals can be downloaded at:
agora.io/developer

Known Issue and Limitation

- Each conference can have no more than one whiteboard instance.
- The users can use the whiteboard function on the Agora supported platforms including Web, Mac and Windows. On iOS and Android, the users have view-only permission.
- If your application is on Mac, the file upload function is not supported.

  This issue is caused by OS X unable to open the panel in WKWebView, and the file upload window cannot popup.

  Wait for OS X to update.

Getting Started

Where to Get the SDK

The Agora Native SDK for Android is available from agora.io/developer, or contact sales@agora.io.

Component	Description
./doc	Agora Whiteboard SDK Documentation (Both English and Chinese): Agora_Whiteboard_SDK_Reference_Manual_v1_0_EN.pdf Agora_Whiteboard_SDK_Reference_Manual_v1_0_CHS.pdf
./client	Agora Whiteboard JavaScript library and web demo application
./server	Web server-side sample code for Channel Key generation

About Keys

This section describes the concepts and use of vendor, sign, and Channel Key.
Each Agora account can create multiple projects, and each project has a unique App ID. Be sure to obtain a **App ID**, which is required when you use APIs to access the Agora Global Network. In the Agora network, you are isolated from others by App ID. There is no overlap even when channel have the same name.
App ID is a static key, if someone else illicitly obtained your static App ID they can use it on their own Agora SDK client applications. If they find the channel names of your organization, they can even interfere with your communication sessions.

A Channel Key is a more secure user authentication schema for the Agora SDK. Whenever a user tries to enter a channel to access the Agora service, the back-end services use an App ID and an App Certificate to generate a new Channel Key based on the HMAC encryption algorithm. The Channel Key is then passed to the client application. The client application calls the Agora.Whiteboard.join(for web application) or use WebView to access the Agora Whiteboard URL(for non-web application) and passes the encoded Channel Key to the Agora server for the user authentication.

Note
Agora Whiteboard SDK does not support the recording function currently.

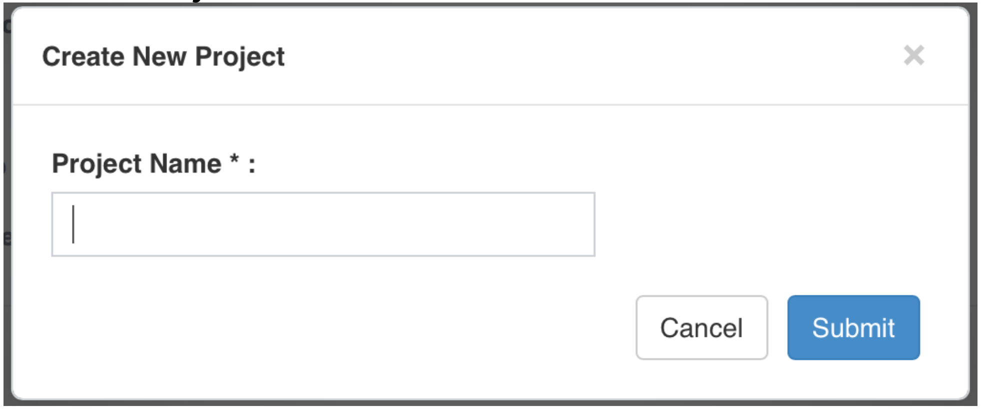
Obtaining and Using an App ID

Obtaining an App ID

Each Agora account can create multiple projects, and each project has a unique App ID.

1. Sign up for a new account at https://dashboard.agora.io/.

2. Click **Add New Project** on the **Projects** page of the dashboard.
3. Fill in the **Project Name** and click **Submit**.



4. Find your App ID under the project that you have created.

Using an App ID

Access the Agora services by using your unique App ID :

Enter the App ID in the web demo application (only for web-application).

Set the **key** parameter to the value of the App ID when joining a session.

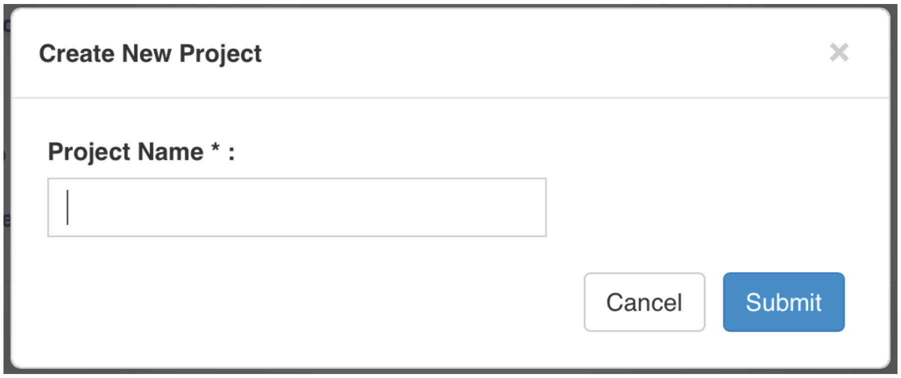
Obtaining and Using a Channel Key

Obtaining an App ID and an App Certificate

Each Agora account can create multiple projects, and each project has a unique App ID and App Certificate.

Follow the steps below to create a project to obtain an App ID and an App Certificate at the same time:

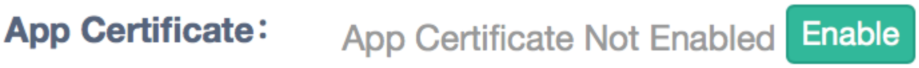
1. Log in to https://dashboard.agora.io.
2. Click **Add New Project** on the **Projects** page of the dashboard.
3. Fill in the **Project Name**, and click **Submit**.



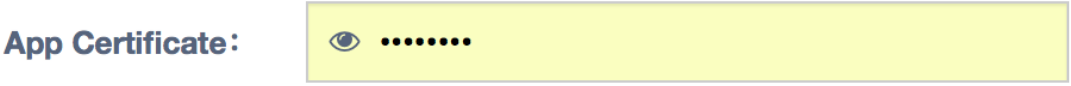
4. Enable the App Certificate for the project.

   a. Click **Edit** on the top right of the project.

   b. Click **Enable** to the right of App Certificate. Read the description **About App Certificate** carefully before you confirm the operation.



   c. Click the “eye” icon to view the App Certificate. You can re-click this icon to hide the App Certificate.



Notes

- If you want to renew an App Certificate, contact support@agora.io.
- Keep your App Certificate on the server and never on any client machine.
- See the table in Increased Security with Channel Key for App Certificate use.

Implementing the Channel Key Scheme

Integrate the Channel Key scheme into your organization’s signaling service. Agora.io provides sample server-side code covering the following languages: Java, C++, Python, node.js, and so on, which you can use directly in your application. For the sample code, refer to the following site: https://github.com/AgoraLab/AgoraDynamicKey

Using a Channel Key

Before a user joins a session to access the Agora service:

1. The client application requests authentication from the signaling server of your organization.
2. The server, upon receiving the request, uses the algorithm provided by Agora.io to generate a Channel Key, then passes the Channel Key back down to the client application.
3. The Channel Key is based on the App Certificate, App ID, Channel Name, Current Timestamp, Client User ID, Lifespan Timestamp, and so on.
4. Set the *Key* parameter to the value of the Channel Key when the client application joins a session.
5. The Agora server receives the Channel Key and confirms that the call comes from a legal user, and then allows it to access the Agora Global Network.

Increased Security with Channel Key

If your organization chooses to use Channel Key, before a user joins a channel, the client application must provide a new Channel Key. The signaling server verifies each user identity. The Channel Key uses the HMAC/SHA1 signature schema to increase security for communications within your organization.

Channel Key Structure

The table below outlines the structure of the Channel Key. Connect all fields in the sequence shown.

Field	Type	Length	Description
Version	String	3	Channel Key version information
Sign	String	40	Hex code for the signature. A 40-byte string calculated by the HMAC algorithm based on inputs including the App Certificate and the following fields: • Service Type: Visible ASCII string provided by Agora.io. See Service Type. • App ID: 32-character App ID string. • Timestamp: The timestamp created when the Channel Key is generated. • Random Number: A 32-bit integer in hex code; generated upon each query. • Channel: The channel name specified by the user, maximum length: 64-bytes. • User ID: The User ID defined by the client. • Call Expiration Timestamp: The timestamp indicates that from the specific moment the user cannot communicate in the channel any more.
App ID	String	32	App ID
Authorized Timestamp	Number	10	The timestamp represented by the number of seconds elapsed since 1-1-1970. The user can use the Channel Key to join a channel within 5 minutes after the Channel Key is generated. If the user does not join a channel after 5 minutes, this Channel Key is no longer valid.
Random Number	Integer	8	A 32-bit integer in hex code; generated upon each query.
Call Expiration Timestamp	Number	10	Set the value to 0 for no limitation to the time of termination. Indicates the exact time when a party can no longer use the Agora service (for example, when someone is forced to leave an ongoing call). [1]

---

[1] When the value is set for Call Expiration Timestamp, it does not mean Channel Key will be expired, but only means the user will be kicked out from the channel. For example, if a conference call is around 1 hour, but if after the call is closed, someone is still in the channel, you will still be charged. By setting this timestamp, the users who have not left the channel after a call is closed will automatically log out.

Service Type

Service	Value	Description
Session	ACS	The audio and video services provided by Agora.io. For example, when you call the joinChannel API, and a Channel Key is required, use this service type to generate the Channel Key.

Sign encryption algorithm： HMAC/SHA1

The Channel Key encoding uses the industry standard HMAC/SHA1 approach for which there are libraries on most common server-side development platforms, such as Node.js, PHP, Python, Ruby and others. For more information, refer to: http://en.wikipedia.org/wiki/Hash-based_message_authentication_code

Realizing Whiteboard Function – Web Application

Deploying Your Web Application

Applications using the Agora Whiteboard SDK are standard JavaScript applications. To deploy the application, you need to load the Agora JS library and also need access to the JS extension libraries provided with the SDK.

Follow your normal JavaScript hosting procedures when deploying the JavaScript libraries:

1. Load the Agora JS library:

   AgoraWBSDK-1.0.0.js

2. JS extension libraries are also needed:

   vendor-bundle.js

Running the Demo Application

Client

The default sample code only requires your static App ID (which is entered into the demo application web page).

To run the provided demo application:

1. Ensure that you have a local web server installed, such as Apache, NginX, or Node.js.
2. Deploy the files under ./client/ to your web server, set the Key value in the file script-host.js and script-guest.js to your App ID, and then launch your http/https service.
3. Access the demo application page on your web server using one of the browsers listed in Section [Supported Web Browsers](#).

Do the following if you want to run the demo application with a Channel Key:
Note: For more information, see Obtaining and Using a Channel Key.

1. Set up and launch the key-generation server.
2. Make an http or https client request for the key-generation server to get a new Channel Key.
3. In script-host.js and script-guest.js (both under client/js/), replace the Key with the newly generated Channel Key before calling Agora.Whiteboard.join.

Server

This code is only needed if you want to experiment with using the more secure Channel Key. In production environment, you integrate this logic into your own server-side applications and (re)code this in the programming languages you are already using for your server-based functionality.

The sample code is in JavaScript and requires a standard Node.js server:
1. Install a standard Node.js server within your server or cloud infrastructure.
2. Run ‘npm install’ under ../server/nodejs/.
3. Fill in the values of your APP_ID and APP_Certificate in
../server/nodejs/DemoServer.js.
4. Launch the server with ‘node DemoServer.js’.

Realizing the Whiteboard Function – Non-Web Application

Non-Web application (on any of the following platforms, iOS, Android, Mac or Windows) developers can use WebView to access the Agora Whiteboard URL service:

- **URL:** https://wb.agora.io/html/agora-wb.html
- **Demo Usage:** https://wb.agora.io/html/agora-wb.html?key=key1&cname=name1&role=host&uinfo=user1

Parameter	Description	Required?
key	Key can be one of the following: - App ID: provided by Agora during registration. - Channel Key: the token generated with App ID and App Certificate. A NodeJS implementation of token-gen algorithm is provided. This is the safest and recommended way to access the Agora Global Network.	Yes
cname	The unique channel name for the Agora whiteboard session.	Yes

Parameter	Description	Required?
role	User Role **host** : the organizer **guest**: the participant The organizer can control the whiteboard for all the activities, but the participant has view-only rights.	Yes
uinfo	The username to identify the user. If not specified, the server will generate a random user name.	No
expire	If the expiration date not specified, the validity period is 24 hours starting from the moment the channel is created.	No
width	The width of the whiteboard, and it is 1024 by default.	No
height	The height of the whiteboard, and it is 768 by default.	No

Agora Whiteboard SDK - API Reference

The Agora Whiteboard SDK library includes the following class:

Agora.Whiteboard	Use the Agora.Whiteboard object to join and leave shared sessions.

join(params)

This method allows user to join a shared session. It will create a session if the user initiates that call.

Parameter	Type	Description
params	Object	The only parameter for the join method, which includes the following settings.

key	String	Key can be one of the following: - App ID: provided by Agora during registration. - Channel Key: the token generated with App ID and App Certificate. A NodeJS implementation of token-gen algorithm is provided. This is the safest and recommended way to access the Agora Global Network.
cname	String	The string that provides the unique channel name for the Agora.Whiteboard session.
host	String	The role of the user: 1: host 0: attendee Host can take control of the whiteboard, attendee is view only.
expire	Number	If the expiration date not specified, the validity period is 24 hours starting from the moment the channel is created.
container	String	The identity of the DOM node.
width	String	The container width.
height	String	The container height

uinfo	String	(optional) user name

Sample code:

```
var hostParams = {
        key       : 'f4637604af81440596a54254d53ade29',
        cname    : 'PES-2017',
        host      : 1,
        width     : 800,
        height    : 600,
        container  : "wbHost"
};
 /* Call AgoraWhiteBoardApi */
Agora.Whiteboard.join(hostParams);
```

leave(url)

This method allows users to leave a shared session.

Parameter Name	Type	Description
url	String	(optional) Specify the url that the user will be redirected to.

Sample Code:

```
Agora.Whiteboard.leave(“http://sample.com”);
```