# LarkXR DataChannel SDK 3.2.3.1

## Brief Description

This SDK enables data interaction functions by integrating with web or mobile clients.The current SDK supports server version 3.2.2.0 and above.

## Type Definitions

|  |  |
| --- | --- |
| XR\_SUCCESS | Success |
| XR\_ERROR\_INTERFACE\_FAILED | Unable to Create Communication Interface |
| XR\_ERROR\_SERVER\_UNSUPPORT | Current Application,Server Does Not Support DataChannel |
| XR\_ERROR\_PARAM | Parameter Error |
| XR\_ERROR\_OPREATION | Operation Error |

## Interface Description

### lr\_client\_register\_taskstatus\_callback

LARKXR\_API void DC\_CALL lr\_client\_register\_taskstatus\_callback(on\_taskstatus taskstatus,void\* user\_data = NULL);

Function Description：

This interface allows the application caller to obtain the connection status of the client and the taskid of the current application.

Parameter Description:

1. on\_taskstatusThe callback function for response notifications.
2. user\_data:A pointer to user data passed in,facilitating callbacks for object-oriented languagesC++/C#.

### lr\_client\_start

LARKXR\_API int DC\_CALL lr\_client\_start(const char\* taskid, on\_connected cb\_connected,on\_data cb\_data,on\_disconnected cb\_disconnected,void\* user\_data);

Function Description:

Create a communication interface.

Parameter Description:

1. taskid:A parameter retained for compatibility with older SDK versions.It can usually be passed as`NULL`.
2. on\_connected: An asynchronous callback function to notify whether the connection to the server is successful.It must not be`NULL`.
3. on\_data:A callback function to receive data from web or other clients.It must not be`NULL`.on\_error: A callback function for connection disconnection and error notifications.It must not be`NULL`.
4. user\_data:A pointer to user data passed in,facilitating callbacks for object-oriented languagesC++/C#.

Return ValueRefer to the type definitions.

### lr\_client\_send

LARKXR\_API int DC\_CALL lr\_client\_send(enum data\_type type, const char\* data, *size\_t* size);

Function Description:

Send data,supporting both string and binary data.

Parameter Description:

1. type: The type of data to be sent,either string or binary data.
2. data: Pointer to the data.
3. size: Length of the datafor strings,it is recommended to include the length of the null terminator\0.

Return Value:

Less than 0 indicates a failure to send.Refer to the type definitions.A return value equal to the length of the data indicates a successful send.

### lr\_client\_stop

LARKXR\_API void DC\_CALL lr\_client\_stop();

Function Description:

Terminate the communication interface.

### lr\_client\_register\_aivoice\_callback

Function Description:

Register the intelligent voice interaction communication callback interface.

Parameter Description：

1. on\_aivoice\_callback:The server pushes the intelligent voice structure.

struct AiVoicePacket

{

bool url; // true: online audio URL (mp3), false: audio pack (pcm)

unsigned int voice\_id; // Voice ID

const char\* online\_url; // If url is true, this field is the URL address; otherwise, this field is NULL

int url\_size; // URL length, including '\0'

const char\* nlg; // Text of the current voice interaction

int nlg\_size; // Length of the interaction text, including '\0'

// If URL is false, the following fields describe each pcm packet

unsigned int slice\_id; // Voice slice ID

int samples\_per\_sec; // e.g., 16000

int channels; // e.g., 1

const char\* audio; // Pointer to the data packet, which is empty if url is true

int size\_byte; // Number of bytes per packet

bool last\_packet; // Whether it is the last packet

};

Demo:

• `c++`Contains code snippets for C++development of the data channel,suitable for C++-based development scenarios.

• `webclient`Includes a web client demo,which is a front-end page with an iframe for testing purposes.It is recommended to use WebSDK in actual production environments.

• `unity`Contains the source code for the Unity test application of the data channel.After packaging,the application can be uploaded directly to the LarkXR backend and used for joint debugging and testing with the web client.

• `unreal`Contains the source code for the Unreal test application of the data channelrefer to theCSDN blog.After packaging,the application can also be uploaded to the LarkXR backend and used for joint debugging and testing with the web client.