

Tencent Cloud AI Digital Human

Client SDK Integration

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Client SDK Integration

 H5 SDK Integration

 HTML5 SDK API Description for Client Rendering

Client SDK Integration

H5 SDK Integration

HTML5 SDK API Description for Client Rendering

Last updated : 2024-11-07 09:51:58

Note:

Current version: 3.5.6

SDK Package File Description

The SDK file TXIVHSDK_Web_Client_Vx.x.x_Release.js.

A folder named unzip (available since version 3.4.0): It contains unzip_worker.js, unzip_wasm.js, and unzip_wasm_bg.wasm and is used for decompressing ZIP files. For details, see [Initializing the Decompression Feature](#).

SDK API/Event Description

setPrivatization Settings

Set it before the init method. After settings, you do not need to pass in the sign parameter when calling the init method. If it is not set, you need to pass in the sign parameter.

Note: It needs to be set in a private deployment environment.

Parameter description

Parameter name	Required	Type	Description
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description

appKey	Yes	String	See Product Introduction to obtain the key. After this parameter is set, you can initiate authentication locally.
accessToken	Yes	String	See Product Introduction to obtain the key. After this parameter is set, you can initiate authentication locally.
httpOrigin	Yes	String	Optional configuration with a high priority. If this parameter is set, all HTTP requests will be sent according to this origin. If it is not set, requests will be sent via the public cloud address.
wsOrigin	Yes	String	Optional configuration with a high priority. If this parameter is set, all socket requests will be sent according to this origin. If it is not set, requests will be sent via the public cloud address.

Return value description

Parameter name	Required	Type	Description
data	Yes	Boolean	true: The setting is successful. false: The setting failed.

Sample code

```
const result = IVH.setPrivatization({
  appKey: 'appkey in the key',
  accessToken: 'accesstoken in the key',
  httpOrigin: 'http://www.test.com',
  wsOrigin: 'ws://www.test.com',
});
console.log(result);
```

Init (Initialization)

Note:

The ZIP file paths can be passed in since version 3.4.0, which can reduce the network file loading time. For details, see [Initializing the Decompression Feature](#). **(ZIP file paths can be passed in only after initUnzip is called.)**

Parameter description

Parameter name	Required	Type	Description
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description
element	Yes	Element/String	Container element object, or selector for finding the container element, for example, "#element ID".
virtualmanProjectId	Yes	String	For details, see Usage Process .
sign	Yes	String	For details, see Product Introduction . If this parameter is not set, appKey and accessToken need to be set in setPrivatization. If no setPrivatization is available and the sign parameter is set, public cloud authentication can be conducted.
modelPath	Yes	String	Set either this parameter or modelData. This parameter has a lower priority than other parameters. Model path. Both relative and absolute paths are supported. Note: ZIP links are supported. For details, see Initializing the Decompression Feature .
modelData	Yes	ArrayBuffer	Set either this parameter or modelPath. This parameter has a higher priority than other parameters. Model data. After fetching data, you can convert it to an ArrayBuffer.
actionPath	Yes	Array	Set either this parameter or actionData. This parameter has a lower priority than other parameters. Action path. Both relative and absolute paths are supported. Note: ZIP links are supported. For details, see Initializing the Decompression Feature . When the ZIP link of an action file is passed in, the first value in the array can be the URL, and other values will be ignored. When the ZIP file is used: Set silent action: Rename the corresponding_action.json file in the ZIP package to default_action_idle.json .

			Set opening action: Rename the corresponding_action.json file in the ZIP package to opening_action_once.json .
actionData	Yes	Array	Set either this parameter or actionPath. This parameter has a higher priority than other parameters. Action data.
defaultActionIdx	Yes	Int	The subscript in the action array, which will be regarded as the silent action. The default value is 0. Note: This parameter does not take effect when an action file in ZIP format is passed in.
openingActionIdx	Yes	Int	The subscript in the action array, which should not be the same as the silent action and should have a valid value. It will be executed once after initialization. Note: This parameter does not take effect when an action file in ZIP format is passed in.
modelOption	Yes	ModelOption	The model rendering parameter.
configPath	Yes	String	Set either this parameter or configData. This parameter has a lower priority than other parameters. Configuration file path. Both relative and absolute paths are supported. Note: If the downloaded model asset package contains a configuration file, you need to load it during initialization.
configData	Yes	Object	Set either this parameter or configPath. This parameter has a higher priority than other parameters. JSON data. After fetching data, you can convert it to a JSON file. Note: If the downloaded model asset package contains a configuration file, you need to load it during initialization.
isIntl	Yes	Boolean	Whether International Edition is used. The default value is false (Chinese Edition).

Warning:
Chinese | International Edition accounts and APIs are not mutually compatible. If they do not match, the API will be abnormal.

ModelOption Type Description

Parameter name	Required	Type	Description
mode	Yes	Array<String>	Mode collection. Note: position': Positioning mode. The position can be adjusted freely after configuration. Example: ['position'].

Sample code

```
// Path loading method.
IVH.init({
  element: '#model',
  virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxx',
  sign: 'xxxxxxxxxxxxxxxxxxxx',
  modelPath: `./model/yangxiaoyun/200.glb`,
  actionPath: [
    `./model/yangxiaoyun/action/hands_drooping.json`,
  ],
  defaultActionIdx: 0
});

// Data loading method.
const option = {
  modelPath: `./model/yangxiaoyun/200.glb`,
  actionPath: [
    `./model/yangxiaoyun/action/welcome.json`,
    `./model/yangxiaoyun/action/thinking.json`,
  ],
};
const fetchArr = option.actionPath.map(n => {
  return fetch(n).then(d => d.json());
});
Promise.all([fetch(option.modelPath).then(d => d.arrayBuffer())].concat(fetchArr)).
  const opt = {
    element: '#model',
```



```

virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxxxx',
sign: 'xxxxxxxxxxxxxxxxxxxxxx',
defaultActionIdx: 0,
};
opt.modelData = data[0];
opt.actionData = [
  {
    name: 'welcome',
    data: data[1],
  },
  {
    name: 'thinking',
    data: data[2],
  }
];
IVH.init(opt);
});

```

Loading Action

Note:

This action should be executed after initialization.

Parameter description

Parameter name	Required	Type	Description
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description
actionPath	Yes	Array	Set either this parameter or actionData. This parameter has a lower priority than other parameters. Action path. Both relative and absolute paths are supported.
actionData	Yes	Array	Set either this parameter or actionPath. This parameter has a higher priority than other parameters. Action data.

Sample code

```
IVH.on('init', e => {
  console.log('init:', e);
  IVH.loadAction({
    actionPath: [
      `./model/yangxiaoyun/action/welcome.json`,
    ],
  });
});
```

Initializing AudioContext

Sample code

Note:

Due to iOS's audio auto-play restrictions, you should call the init method in the callback of the behavior event:

Correct example: Call initAudio at the level of the user behavior event callback function.

```
document.querySelector('#btn').onclick = function (e) {
  IVH.initAudio();
  const txt = document.querySelector('#txt').value.trim();
  if (txt) {
    setTimeout(() => {
      IVH.play(txt);
    }, 1000);
  }
}
```

Incorrect example: Call initAudio at other function levels. In this case, iOS will consider it non-user behavior.

```
document.querySelector('#btn').onclick = function (e) {
  const txt = document.querySelector('#txt').value.trim();
  if (txt) {
    setTimeout(() => {
      IVH.initAudio();
      IVH.play(txt);
    }, 1000);
  }
}
```

Play (Broadcast)

Note:

Execute initAudio before each call.

Parameter description

Parameter name	Required	Type	Description
text	Yes	Object	Text to be played, which supports the SSML specification. (Pure action tags only support single actions.)
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description
isChat	Yes	Boolean	Whether to enable the dialogue mode. The default value is false (plain text-driven). When this parameter is set to true, the corresponding dialogue mode is enabled. The dialogue modes include customer service dialogue and large model dialogue.
isNew	Yes	Boolean	This parameter is valid when isChat is true. Whether the new and old sessions belong to the same session (multi-round). The settings will affect the results of multi-round sessions.
isStream	Yes	Boolean	This parameter is valid when isChat is false. When this parameter is set to true, the streaming text-driven mode is used. The default value is false. Warning: Streaming text does not support the SSML format.
seqNo	Yes	Int	This parameter should be passed when isStream takes effect. Serial number of a streaming text shard, which starts from 1.
isFinal	Yes	Boolean	This parameter should be passed when isStream takes effect. End marker of streaming text, which should be passed in for each streaming text segment.
streamId	Yes	String	This parameter should be passed when isStream takes effect. Request ID of streaming text, which is a 32-character UUID used to identify the same stream.
userId	Yes	String	This parameter is valid when isChat is true. If this parameter is passed in, isNew will be invalid. You can manually identify whether it belongs to the same session (used for multi-terminal queries).

Note:

The value of this parameter is a string of no more than 64 characters, which is passed in by customers during the large model dialogue.

Sample code

```
document.querySelector('#btn').onclick = function (e) {
  const txt = document.querySelector('#txt').value.trim();
  if (txt) {
    IVH.initAudio();
    IVH.play(txt, {
      isChat: true,
    });
  }
}

// Simulated streaming API test case, which is for reference only.
document.querySelector('#btn7').addEventListener('click', e => {
  const arr = [
    'The weather is nice today.',
    'It is sunny and breezy.',
    'We can go out and have fun.'
  ];
  IVH.initAudio();
  const streamId = crypto.randomUUID().replace(/-/g, '');
  for (let i = 0; i < arr.length; i++) {
    IVH.play(arr[i], {
      isStream: true,
      seqNo: i + 1,
      isFinal: i === arr.length - 1,
      streamId,
    })
  }
})
```

Version Information**Returned values**

Parameter name	Required	Type	Description
result	Yes	Object	A JSON object that also contains version information objects.

```
{
  version: '1.0.0'
}
```

Sample code

```
/**
 * @returns Object
 * * @property {String} version: version information, for example, '1.0.0'.
 */
IVH.info()
```

Setting volume level

Parameter description

Parameter name	Required	Type	Description
val	Yes	Number	Volume, which ranges from 0 to 1.

Sample code

```
let volume = 1;
document.querySelector('#btn2').onclick = function (e) {
  volume = 1 - volume;
  IVH.setVolume(volume);
}
```

Interrupting the Broadcast

Sample code

```
document.querySelector('#btn3').onclick = function (e) {
  IVH.stop();
}
```

Obtaining the Instruction Configuration

This method has been supported since version 3.1.0.

Parameter description

Parameter	Required	Type	Description
-----------	----------	------	-------------

name			
key	Yes	String	Instruction key. You can obtain the configured instruction information based on it.

Returned values

Parameter name	Required	Type	Description
result	Yes	Promise<Object>	JSON object returned asynchronously. For details about key fields, see Appendix Description .

Sample code

```
const result = await IVH.getActionConfig('xxx');
console.log(result);
```

Obtaining the Configuration Information

This method has been supported since version 3.2.0.

Returned values

Parameter name	Required	Type	Description
result	Yes	Object	JSON data is returned. It is generally saved as conf.json after the model position is adjusted. During the next initialization, you can load it to adjust the initial model position.

Sample code

```
const result = IVH.getConf();
console.log(result);
```

Resetting the Position Information

This method has been supported since version 3.2.0. If you have customized a position during initialization, you can call this method to reset the position.

Sample code

```
IVH.resetPositionInfo();
```

Destroy

This method has been supported since version 3.3.0. It is used to terminate instances to free up memory.

Note:

After termination, if you want to render again, you need to re-initialize the corresponding method.

Sample code

```
IVH.destroy();
```

Initializing the Decompression Feature

This method has been supported since version 3.4.0. After initialization, the model and action files can be compressed into the ZIP package, reducing network request time.

Note:

Only ZIP files and the Deflated compression algorithm are supported.

iOS 11.0 or above and Android 7.0 or above are supported.

Parameter description

Parameter name	Required	Type	Description
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description
path	Yes	String	The path where files for decompression are located. There are three such files, namely unzip_worker.js, unzip_wasm.js, and unzip_wasm_bg.wasm. Place them in the same folder and pass the path, for example, <code>http://xxx.com/unzip/</code> , or a relative path, such as <code>./unzip/</code> , to the path parameter.

Returned values

Parameter name	Required	Type	Description
result	Yes	Promise<Boolean>	Boolean value returned asynchronously. true indicates

success, and false indicates failure. In case of a failure, the relevant error information will be returned in the error event.

Sample code

```
// Ensure that initUnzip returns true before the ZIP link is passed in.
(async () => {
  const result = await IVH.initUnzip({
    path: './unzip'
  });
  result && IVH.init({
    element: '#xxxx',
    sign: 'xxxxx',
    virtualmanProjectId: 'xxxxx',
    modelPath: './model/model.zip',
    // For the ZIP link of an action file, only the URL in the array is considered,
    actionPath: [
      './model/action.zip',
    ],
  });
})();
```

Loading Component Data

This method has been supported since version 3.5.0. After the init and initUnzip methods are called for a model, component data (for example, matching actions with the prop model) can be loaded.

Parameter description

Parameter name	Required	Type	Description
option	Yes	Option	Customized type

Option Type Description

Parameter name	Required	Type	Description
zipPath	Yes	String	The path where component compression files are located. Due to the large number of files related to prop matching actions, if this parameter is not set, subsequent exceptions may occur. Therefore, file verification will be performed during loading.

			In case of an exception, the relevant exception information will be returned in the error event. In case of a success, the relevant loading information will be returned in the actionLoad event.
--	--	--	---

Sample code

```
// Ensure that both init and initUnzip methods are called.
(async () => {
  const result = await IVH.initUnzip({
    path: './unzip'
  });
  IVH.on('init', e => {
    IVH.loadComponent({
      zipPath: './model/aiyun/aiyun.zip',
    })
  });
  result && IVH.init({
    element: '#xxxx',
    sign: 'xxxxx',
    virtualmanProjectId: 'xxxxx',
    modelPath: './model/model.zip',
    // When the ZIP link of an action file is passed in, the first value in the
    actionPath: [
      './model/action.zip',
    ],
  });
})();
```

Listening for events

Parameter description

Parameter name	Required	Type	Description
event	Yes	String	Event
callback	Yes	Function	Callback function.

List of Listenable Events

Event	Required	Description
error	Yes	Triggered when an error occurs.

init	Yes	Triggered after model initialization is completed.
actionLoad	Yes	Triggered after action data is loaded.
actionEnd	Yes	Triggered after the action is executed.
play	Yes	Triggered after the TTS voice is played.
reskinEnd	Yes	Skin change event, which is triggered after skin change ends or when it is abnormal.
nlp	Yes	Dialogue return result, which is used for rich text display. For details, see Appendix Description .
sentence	Yes	Sentence-based broadcast information in large model mode. seqNo corresponds to seqNo returned in the NLP event. status indicates the playback status, including start and over.

Sample code

```
IVH.on('init', e => {  
  console.log('init:', e);  
});
```

Guide

Switching Between Two Models to Achieve the Dressing Effect

Sample code

```
// Initialize clothing model A first.  
IVH.init({  
  element: '#model',  
  virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxx',  
  sign: 'xxxxxxxxxxxxxxxxxxxx',  
  modelPath: `./model/modelA.glb`,  
  actionPath: [  
    `./model/actionA/xxxx.json`,  
  ]  
});  
  
// Terminate the previous clothing model A before the switch.  
IVH.destroy();
```

```
// Re-initialize clothing model B.
IVH.init({
  element: '#model',
  virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxx',
  sign: 'xxxxxxxxxxxxxxxxxxxx',
  modelPath: `./model/modelB.glb`,
  actionPath: [
    `./model/actionB/xxxx.json`,
  ]
});
```

Adjusting the Starting Position and Size of the Model

Sample code

```
// 1. Initialize the model and enable the positioning mode.
IVH.init({
  element: '#model',
  virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxx',
  sign: 'xxxxxxxxxxxxxxxxxxxx',
  modelPath: `./model/model.glb`,
  actionPath: [
    `./model/action/xxxx.json`,
  ],
  modelOption: {
    mode: ['position']
  }
});

// 2. Scroll the mouse wheel to zoom in or out the model size, and right-click to d
getConf.addEventListener('click', e => {
  const blob = new Blob([JSON.stringify(IVH.getConf())], { type: 'application/json'
  // Name of the downloaded file.
  const filename = 'conf.json';
  // Create a hidden downloadable link.
  const eleLink = document.createElement('a');
  eleLink.download = filename;
  eleLink.style.display = 'none';
  // Convert the download content into a blob URL.
  eleLink.href = URL.createObjectURL(blob);
  // Trigger the click action.
  document.body.appendChild(eleLink);
  eleLink.click();
  // Remove the URL.
  document.body.removeChild(eleLink);
  URL.revokeObjectURL(eleLink.href);
```

```
});

// 3. Re-initialize the model and load configPath/configData. The model will be ren
IVH.init({
  element: '#model',
  virtualmanProjectId: 'xxxxxxxxxxxxxxxxxxxx',
  sign: 'xxxxxxxxxxxxxxxxxxxx',
  modelPath: './model/model.glb',
  actionPath: [
    './model/action/xxxx.json',
  ],
  configPath: './model/conf.json',
});
```

Appendix Description

Model

The following table describes the key fields returned by the NLP event.

Attribute Name	Description
replyType	cloudAiGpt: Tencent Cloud large model dialogue. yunxiaowei: Tencent Cloud Xiaowei customer service. cloudAiWaiting: Script for waiting for the first package due to timeout. cloudAiTimeOut: Script for timeout without response and the session ends. sensitive: Fixed script for sensitive content. input: Content of the plain or streaming text input. enhanceText: Plain text-driven content matches the content in script management.
cloudAiExtra	String in JSON text format, which is an additional supplementary field for large models. This field has a value when replyType is cloudAiGpt and isFinal is true.
record_id	Message ID, which is the attribute of cloudAiExtra.
-- references	Answer source. For details.
seqNo	Serial number of a clause.
isFinal	Whether it is the last sentence.
isHighLight	Highlight tag.
contentType	Type in MD format. 0: unknown. 1: ordinary string.

- 2: ordered list.
- 3: unordered list.
- 4: image link.
- 5: HTTP link.
- 6: table.
- 7: code block.

Customer Service Dialogue

The following table describes the key fields returned by the NLP event. (These fields have been standardized to start with a lowercase letter since version 2.3.1.)

Attribute Name	Description
replyDisplay	Content configured at the configuration end, which uses the HTML syntax.
replyPro	TTS broadcast content, which uses the SSML syntax.
interactionType	Special message type. For details, see Description of Special Messages below.
interactionContent	Special message content. For details, see Description of Special Messages.

Description of Special Messages

Through the script management module of Tencent Cloud AI Digital Human (TCADH), you can configure special message types for different responses, assisting the integration party in quickly configuring special message content. The following are examples of codes for implementing front-end special message types. The SDK itself does not include front-end interaction styles.

Note:

The SDK itself does not include front-end interaction styles. If you need to use special message types, the integration party needs to develop the code.

Multiple Choice

```
/*
 * Note: Style 1: style=1; style 2: style=2; style 3: style=3;
 */
{
  InteractionContent: "{\\"style\\":1,\\"options\\":[\\"Option 1\\",\\"Option 2\\",
  InteractionType: "OptionInfo"
}
```

Image

```
{
  InteractionContent: "{\\"url\\":\\"https://virtualhuman-cos-test-1251316161.cos.a
  InteractionType: "Image"
}
```

Pop-up window

```
{
  InteractionContent: "{\\"title\\":\\"Pop-up headline\\",\\"content\\":\\"Pop-up c
  InteractionType: "Popup"
}
```

Video

```
{
  InteractionContent: "{\\"url\\":\\"https://virtualhuman-cos-test-1251316161.cos.a
  InteractionType: "Video"
}
```

Image + Multiple Choice

```
/*
 * Note: Style 1: style=1; style 2: style=2; style 3: style=3;
 */
{
  InteractionContent: "{\\"url\\":\\"https://virtualhuman-cos-test-1251316161.cos.ap
  InteractionType: "ImageOption"
}
```

Custom

```
{
  InteractionContent: "{\\"name\\":\\"Headline\\",\\"contents\\":[\\"Content\\" ]}",
  InteractionType: "Customize"
}
```