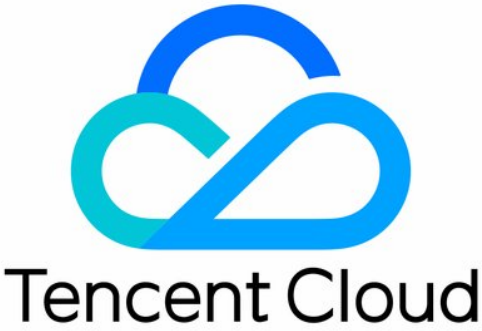
Tencent Cloud AI Digital Human

Server API Integration

Product Documentation



©2013-2025 Tencent Cloud International Pte. Ltd.

Copyright Notice

©2013-2025 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

All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. 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

Server API Integration
- Avatar aPaas API Calling Methods
- Avatar Image Customization and Voice Clone API Documentation
  - Overview
  - Avatar Customization
    - API Calling Logic Diagram
    - Obtaining the Temporary Upload Token
    - Uploading Materials to Tencent Cloud COS
    - Customization API
    - Progress Query Interface
    - Effect Confirmation API
  - Voice Clone (Basic Edition)
    - API Calling Logic Diagram
    - Audio Quality Inspection Task Creation API
    - Audio Quality Inspection Task Status Query API
    - Obtaining the Temporary Upload Token
    - Uploading Materials to Tencent Cloud COS.
    - Customization API
    - Progress Query API
    - Effect Confirmation API
  - Voice Clone (Ultra Edition)
    - API Calling Logic Diagram
    - Obtaining the Voice Training Text
    - Obtaining the Temporary Upload Token
    - Uploading Materials to Tencent Cloud COS
    - Submitting Audio Quality Inspection Task
    - Querying Audio Quality Inspection Task Progress
    - Customization API
    - Progress Query API
  - Result Code Description
  - Interface Integration FAQs
- Video Generation Service API Documentation
  - Overview
  - Digital Human aPaaS API Calling Methods
  - Audio Production API

©2013-2025 Tencent Cloud International Pte. Ltd.

- Video Production API - Basic Edition
- Audio and Video Production Progress Query API
- Video Production API - Advanced Version
- Customer Resource Query Anchor API
- Querying All Images of a Specific Anchor
- Querying the Supported Timbres for VirtualmanKey (to Be Deprecated)
- Querying the Supported Actions for VirtualmanKey
- Appendix
  - Appendix I: Result Code Dictionary
  - Appendix II: Callback Request Body Format
- API Integration FAQs

Interactive Digital Human Service API Documentation

- Overview
- Digital Human aPaaS API Calling Methods
- Live Session Management
  - Creating a New Live Stream Session
  - Using the Personal Asset Image to Create a Stream
  - Querying the Session Status
  - Starting Session
  - Closing Live Session
  - Querying the Conversation List under the Digital Human Project
  - Querying the Conversion List under the Personal Asset Image
  - Querying the Session List under UIN
- Instruction-driven
  - Instruction Sending Requirements
  - Creating a Long Connection Channel
  - Text-driven Instructions
  - Voice-driven Instructions
  - Streaming Text-driven Instructions
  - Long Link Downstream Message
  - Heartbeat Instruction
- Other APIs
  - Real-time Update of Digital Human Visual Parameters
  - Text-driven Instructions
  - Obtaining the Key to the TRTC Room
- Error Code Description
- Best Practices demo

Personal Asset Management API Documentation

©2013-2025 Tencent Cloud International Pte. Ltd.

Overview
Digital Human aPaaS API Calling Methods
Querying for Avatar List by Pagination API
Querying Supported Timbres for Avatars (to be Deprecated)
Querying Customer Service Asset Information
Querying Timbre Lists by Pagination
Querying Image Asset Information - Query Anchor
Querying Image Asset Information - Querying all Avatars under the Anchor
Querying the List Of Actions Supported by the Avatar
Appendix 1 - Service Asset Type
Appendix 2 - Emotional Style
Appendix 3 - Digital Human Type
Appendix 4 - Language List
API Integration FAQs

©2013-2025 Tencent Cloud International Pte. Ltd.

Server API Integration
Avatar aPaas API Calling Methods

Last updated：2024-07-18 17:37:46

This document mainly provides the method to generate the URL for calling the TCADH aPaas platform API, along with sample demos in common programming languages, including Golang, Java, C++, Python, and JavaScript.

I. Obtaining Required Parameters

For obtaining necessary parameters such as appkey and accesstoken, see the methods provided in the corresponding API documentation.

II. Common Parameters

Parameters	Type	Mandatory	Description
appkey	string	Yes	The appkey obtained according to the methods provided in each API documentation.
timestamp	string	Yes	Request timestamp in seconds. The timestamp cannot differ from the current time by more than five minutes, otherwise, authentication will fail.
requestid	string	No	Only some APIs (such as creating a long connection channel for interactive Avatar) require this parameter. Please see the corresponding API documentation for the acquisition method.
signature	string	Yes	Request signature (see Signature Method for details).

III. Signature Method

When calling any aPaas API, you need to include common parameters such as the request signature in the URL as a QueryString.

Request Parameter Signature Steps are as follows:

1. The signature rules are as follows and described with an example (a pseudo-code for reference only):

```
appkey = example_appkey
```

```
accesstoken = example_accesstoken
Domain name routing = https://api.example.com/v2/ivh/example_uri
```

2. According to the corresponding API documentation, all required parameters except for signature should be sorted alphabetically to form the plaintext for generating the signature. The example only uses appkey and timestamp as parameters (please confirm the required parameters for each API in the relevant documentation). The concatenated and sorted string example is:

```
appkey=example_appkey&timestamp=1717639699
```

3. Use the accesstoken to perform HmacSha256 encryption on the signature string, and then encode it with base64:

```
hashBytes = HmacSha256("appkey=example_appkey&timestamp=1717639699","example_access
signature = Base64Encode(hashBytes)
```

4. The obtained signature value is:

```
aCNWYzZdplxWVo+JsqzZc9+J9XrwWWITfX3eQpsLVno=
```

5. Urlencode the signature value (this is mandatory, otherwise it may cause authentication failures), and then concatenate it to get the final request URL as follows:

```
https://api.example.com/v2/ivh/example_uri?appkey=example_appkey&timestamp=17176396
```

IV. Demo

The user only needs to fill in all common parameters required for accessing the corresponding API except the signature into the corresponding positions in the demo to generate the request signature and concatenate it into the complete URL for accessing the API.

Golang

Java

C++

Python

JavaScript

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/base64"
    "fmt"
    "net/url"
```

```
    "sort"
    "strconv"
    "time"
)

func GenSignature(signingContent string, accessToken string) string {

    // Calculate the HMAC-SHA256 value.
    h := hmac.New(sha256.New, []byte(accessToken))
    h.Write([]byte(signingContent))

    // Encode the HMAC-SHA256 value in Base64.
    hashInBase64 := base64.StdEncoding.EncodeToString(h.Sum(nil))

    // URL encode
    encodeSign := url.QueryEscape(hashInBase64)

    // Concatenate the signature.
    signature := "&signature=" + encodeSign

    return signature
}

func GenReqURL(parameter map[string]string, accessToken string, baseURL string) str
    // Concatenate the string to be signed in alphabetical order.
    signingContent := ""
    pathKey := make([]string, 0, len(parameter))
    for k := range parameter {
       pathKey = append(pathKey, k)
    }
    sort.Strings(pathKey)
    for _, k := range pathKey {
       if signingContent != "" {
          signingContent += "&"
       }
       signingContent += k + "=" + parameter[k]
    }

    // Calculate the signature.
    signature := GenSignature(signingContent, accessToken)

    // Concatenate the complete URL for accessing the API.
    return baseURL + "?" + signingContent + signature
}

func main() {
    baseUrl := "https://api.example.com/v2/ivh/example_uri"
```

```
    accesstoken := "example_accesstoken"
    wssUrl := "wss://api.example.com/v2/ws/ivh/example_uri"
    // Example I (Accessing an API that requires appkey and timestamp parameters):
    // User fills in the required common parameters required to generate the signat
    parameter := map[string]string{
        "appkey":    "example_appkey",
        /// The timestamp provided by the user should be in seconds, and the gap bet
        // Example timestamp:
        // "timestamp": "1717639699",
        // It is recommended to use the following statement to generate the current
        "timestamp": strconv.FormatInt(time.Now().Unix(), 10),
    }
    url1 := GenReqURL(parameter, accesstoken, baseUrl)
    // The output with the example timestamp should be as follows:
    // Example 1:https://api.example.com/v2/ivh/example_uri?appkey=example_appkey&t
    fmt.Println("Example 1:" + url1)

    /// Example II (Accessing an API that requires appkey, requestID and timestamp
    // User fills in the required common parameters required to generate the signat
    parameter = map[string]string{
        "appkey":    "example_appkey",
        "requestid": "example_requestid",
        /// The timestamp provided by the user should be in seconds, and the gap bet
        // Example timestamp:
        // "timestamp": "1717639699",
        // The timestamp provided by the user should be in seconds, and the gap betw
        "timestamp": strconv.FormatInt(time.Now().Unix(), 10),
    }
    url2 := GenReqURL(parameter, accesstoken, wssUrl)
    // The output with the example timestamp should be as follows:
    // Example 2:wss://api.example.com/v2/ws/ivh/example_uri?appkey=example_appkey&
    fmt.Println("Example 2:" + url2)
}
```

```
import java.util.*;
import java.util.stream.Collectors;
import java.time.Instant;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;
import java.security.InvalidKeyException;
```

```
import java.security.NoSuchAlgorithmException;
import java.io.UnsupportedEncodingException;

public class Presigned {

    public static String GenSignature(String signingContent, String accessToken) {
        try {
            // Calculate the HMAC-SHA256 value.
            Mac sha256_HMAC = Mac.getInstance("HmacSHA256");
            SecretKeySpec secret_key = new SecretKeySpec(accessToken.getBytes(Stand
            sha256_HMAC.init(secret_key);

            String hashInBase64 = Base64.getEncoder().encodeToString(sha256_HMAC.do

            // URL encode
            String encodeSign = URLEncoder.encode(hashInBase64, StandardCharsets.UT

            // Concatenate the signature.
            String signature = "&signature=" + encodeSign;

            return signature;
        } catch (NoSuchAlgorithmException | InvalidKeyException | UnsupportedEncodi
            e.printStackTrace();
            return null;
        }
    }

    public static String GenReqURL(Map<String, String> parameter, String accessToke
        // Concatenate the string to be signed in alphabetical order.
        String signingContent = parameter.entrySet().stream()
                .sorted(Map.Entry.comparingByKey())
                .map(entry -> entry.getKey() + "=" + entry.getValue())
                .collect(Collectors.joining("&"));

        // Calculate the signature.
        String signature = GenSignature(signingContent, accessToken);

        // Concatenate the complete URL for accessing the API.
        return baseURL + "?" + signingContent + signature;
    }

    public static void main(String[] args) {
        String baseUrl = "https://api.example.com/v2/ivh/example_uri";
        String accesstoken = "example_accesstoken";
        String wssUrl = "wss://api.example.com/v2/ws/ivh/example_uri";

        // Example I (Accessing an API that requires appkey and timestamp parameter
```

```
        // User fills in the required common parameters required to generate the si
        Map<String, String> parameter = new TreeMap<>();
        parameter.put("appkey", "example_appkey");
        /// The timestamp provided by the user should be in seconds, and the gap be
        // Example timestamp:
        // parameter.put("timestamp", "1717639699");
        // It is recommended to use the following statement to generate the current
        parameter.put("timestamp", String.valueOf(Instant.now().getEpochSecond()));
        String url = GenReqURL(parameter, accesstoken, baseUrl);
        // The output with the example timestamp should be as follows:
        // Example 1:https://api.example.com/v2/ivh/example_uri?appkey=example_appk
        System.out.println("Example 1:" + url);

        /// Example II (Accessing an API that requires appkey, requestID and timest
        parameter.clear();
        parameter.put("appkey", "example_appkey");
        parameter.put("requestid", "example_requestid");
        /// The timestamp provided by the user should be in seconds, and the gap be
        // Example timestamp:
        // parameter.put("timestamp", "1717639699");
        // It is recommended to use the following statement to generate the current
        parameter.put("timestamp", String.valueOf(Instant.now().getEpochSecond()));
        url = GenReqURL(parameter, accesstoken, wssUrl);
        // The output with the example timestamp should be as follows:
        // Example 2:wss://api.example.com/v2/ws/ivh/example_uri?appkey=example_app
        System.out.println("Example 2:" + url);
    }
}
```

```
#include<iostream>
#include <map>
#include<string>
#include<vector>
#include<algorithm>
#include <ctime>
#include <iomanip>
#include <sstream>
#include<openssl/hmac.h>
#include<openssl/sha.h>

using namespace std;

static const string base64_chars =
```

```
              "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
              "abcdefghijklmnopqrstuvwxyz"
              "0123456789+/";

string url_encode(const string &value) {
    ostringstream escaped;
    escaped.fill('0');
    escaped<< hex;


    for (size_t i = 0; i< value.length(); ++i) {
        char c = value[i];
        if (isalnum(c) || c == '-' || c == '_' || c == '.' || c == '~') {
            escaped << c;
        } else {
            escaped<< uppercase;
            escaped << '%'<< setw(2)<< int((unsigned char) c);
            escaped<< nouppercase;
        }
    }


    return escaped.str();
}


string base64_encode(unsigned char const* bytes_to_encode, unsigned int in_len) {
    string ret;
    int i = 0;
    int j = 0;
    unsigned char char_array_3[3];
    unsigned char char_array_4[4];
    while (in_len--) {
        char_array_3[i++] = *(bytes_to_encode++);
        if (i == 3) {
            char_array_4[0] = (char_array_3[0] & 0xfc) >> 2;
            char_array_4[1] = ((char_array_3[0] & 0x03) << 4) + ((char_array_3[1] &
            char_array_4[2] = ((char_array_3[1] & 0x0f) << 2) + ((char_array_3[2] &
            char_array_4[3] = char_array_3[2] & 0x3f;


            for(i = 0; (i <4) ; i++)
                ret += base64_chars[char_array_4[i]];
            i = 0;
        }
    }
    if (i)
```

```
    {
        for(j = i; j < 3; j++)
            char_array_3[j] = '\\0';
        char_array_4[0] = (char_array_3[0] & 0xfc) >> 2;
        char_array_4[1] = ((char_array_3[0] & 0x03) << 4) + ((char_array_3[1] & 0xf
        char_array_4[2] = ((char_array_3[1] & 0x0f) << 2) + ((char_array_3[2] & 0xc
        char_array_4[3] = char_array_3[2] & 0x3f;
        for (j = 0; (j < i + 1); j++)
            ret += base64_chars[char_array_4[j]];
        while((i++ < 3))
            ret += '=';
    }
    return ret;
}


string GenSignature(const string& signingContent, const string& accessToken) {


    // Calculate the HMAC-SHA256 value.
    unsigned char hash[EVP_MAX_MD_SIZE];
    unsigned int hashLength;
    HMAC(EVP_sha256(), accessToken.c_str(), accessToken.length(), (unsigned char *)


    // Encode the HMAC-SHA256 value in Base64.
    string hashInBase64 = base64_encode(hash, hashLength);


    // URL encode
    string encodeSign = url_encode(hashInBase64);


    // Concatenate the signature.
    string signature = "&signature=" + encodeSign;


    return signature;
}


string GenReqURL(const map<string, string>& parameter, const string& accessToken, c
    // Concatenate the string to be signed in alphabetical order.
    string signingContent;
    vector<string> pathKey;
    for (const auto& p : parameter) {
        pathKey.push_back(p.first);
```

```
    }
    sort(pathKey.begin(), pathKey.end());
    for (const auto& k : pathKey) {
        if (!signingContent.empty()) {
            signingContent += "&";
        }
        signingContent += k + "=" + parameter.at(k);
    }


    // Calculate the signature.
    string signature = GenSignature(signingContent, accessToken);


    // Concatenate the complete URL for accessing the API.
    return baseURL + "?" + signingContent + signature;
}


// During compilation, linking to the OpenSSL library is required. You can use the
// g++ presigned.cpp -o presigned -lcrypto


int main() {
    string baseUrl = "https://api.example.com/v2/ivh/example_uri";
    string accesstoken = "example_accesstoken";
    string wssUrl = "wss://api.example.com/v2/ws/ivh/example_uri";


    // Example I (Accessing an API that requires appkey and timestamp parameters):
    // User fills in the required common parameters required to generate the signat
    map<string, string> parameter1 = {
        {"appkey", "example_appkey"},
        /// The timestamp provided by the user should be in seconds, and the gap be
        // Example timestamp:
        // {"timestamp" , "1717639699"},
        // It is recommended to use the following statement to generate the current
        {"timestamp", to_string(time(NULL))}
    };
    string url = GenReqURL(parameter1, accesstoken, baseUrl);
    // The output with the example timestamp should be as follows:
    // Example 1:https://api.example.com/v2/ivh/example_uri?appkey=example_appkey&t
    cout << "Example 1:"<< url<< endl;


    /// Example II (Accessing an API that requires appkey, requestID and timestamp
    // User fills in the required common parameters required to generate the signat
```

```
    map<string, string> parameter2 = {
        {"appkey", "example_appkey"},
        {"requestid", "example_requestid"},
        /// The timestamp provided by the user should be in seconds, and the gap be
        // Example timestamp:
        // {"timestamp" , "1717639699"},
        // It is recommended to use the following statement to generate the current
        {"timestamp", to_string(time(NULL))}
    };
    url = GenReqURL(parameter2, accesstoken, wssUrl);
    // The output with the example timestamp should be as follows:
    // Example 2:wss://api.example.com/v2/ws/ivh/example_uri?appkey=example_appkey&
    cout << "Example 2:"<< url<< endl;


    return 0;
}
```

```
import hmac
import hashlib
import time
import base64
from urllib.parse import quote


# Users can generate the timestamp within the function by providing appkey and acce
def GenSignature(signing_content, access_token):


    # Calculate the HMAC-SHA256 value.
    h = hmac.new(access_token.encode(), signing_content.encode(), hashlib.sha256)


    # Encode the HMAC-SHA256 value in Base64.
    hash_in_base64 = base64.b64encode(h.digest()).decode()


    # URL encode
    encode_sign = quote(hash_in_base64)


    # Concatenate the signature.
    signature = f"&signature={encode_sign}"
```

```
    return signature


def GenReqURL(parameter, access_token, base_url):
    # Concatenate the string to be signed in alphabetical order.
    signing_content = '&'.join(f'{k}={parameter[k]}' for k in sorted(parameter.keys


    # Calculate the signature.
    signature = GenSignature(signing_content, access_token)


    # Concatenate the complete URL for accessing the API.
    return f'{base_url}?{signing_content}{signature}'




def main():
    base_url = 'https://api.example.com/v2/ivh/example_uri'
    access_token = 'example_accesstoken'
    wss_url = 'wss://api.example.com/v2/ws/ivh/example_uri'


    # Example I (Accessing an API that requires appkey and timestamp parameters):
    # User fills in the required common parameters required to generate the signatu
    parameter1 = {
        'appkey': 'example_appkey',
        # The timestamp provided by the user should be in seconds, and the gap betw
        # Example timestamp:
        # 'timestamp': '1717639699'
        # It is recommended to use the following statement to generate the current
        'timestamp': int(time.time())  # Use the current timestamp (unit: second)
    }
    url1 = GenReqURL(parameter1, access_token, base_url)
    # The output with the example timestamp should be as follows:
    # Example 1:https://api.example.com/v2/ivh/example_uri?appkey=example_appkey&ti
    print('Example 1:', url1)


    # Example II (Accessing an API that requires appkey, requestID and timestamp pa
    # User fills in the required common parameters required to generate the signatu
    parameter2 = {
        'appkey': 'example_appkey',
        'requestid': 'example_requestid',
```

```
        # The timestamp provided by the user should be in seconds, and the gap betw
        # Example timestamp:
        # 'timestamp': '1717639699'
        # It is recommended to use the following statement to generate the current
        'timestamp': int(time.time())  # Use the current timestamp (unit: second)
    }
    url2 = GenReqURL(parameter2, access_token, wss_url)
    # The output with the example timestamp should be as follows:
    # Example 2:wss://api.example.com/v2/ws/ivh/example_uri?appkey=example_appkey&r
    print('Example 2:', url2)


if __name__ == '__main__':
    main()
```

```
const crypto = require('crypto');

// Users can generate the timestamp within the function by providing appkey and acc
function GenSignature(signingContent, accessToken) {
  // Calculate the HMAC-SHA256 value.
  const hmac = crypto.createHmac('sha256', accessToken);
  hmac.update(signingContent);


  // Encode the HMAC-SHA256 value in Base64.
  const hashInBase64 = hmac.digest('base64');


  // URL encode
  const encodeSign = encodeURIComponent(hashInBase64);


  // Concatenate the signature.
  const signature = `&signature=${encodeSign}`;


  return signature;
}


function GenReqURL(parameter, accessToken, baseURL) {
    // Concatenate the string to be signed in alphabetical order.
    let signingContent = '';
    const pathKey = Object.keys(parameter).sort();
    for (const k of pathKey) {
```

```
      if (signingContent !== '') {
        signingContent += '&';
      }
      signingContent += `${k}=${parameter[k]}`;
    }

    // Calculate the signature.
    const signature = GenSignature(signingContent, accessToken);

    // Concatenate the complete URL for accessing the API.
    return `${baseURL}?${signingContent}${signature}`;
  }




function main() {
  const baseUrl = 'https://api.example.com/v2/ivh/example_uri';
  const accesstoken = 'example_accesstoken';
  const wssUrl = 'wss://api.example.com/v2/ws/ivh/example_uri';


  // Example I (Accessing an API that requires appkey and timestamp parameters):
  // User fills in the required common parameters required to generate the signatur
  const parameter1 = {
      appkey: 'example_appkey',
      /// The timestamp provided by the user should be in seconds, and the gap betw
      // Example timestamp:
      // timestamp: '1717639699',
      // It is recommended to use the following statement to generate the current t
      timestamp: Math.floor(Date.now() / 1000), // Use the current timestamp (unit:
  };
  const url1 = GenReqURL(parameter1, accesstoken, baseUrl);
  // The output with the example timestamp should be as follows:
  // Example 1:https://api.example.com/v2/ivh/example_uri?appkey=example_appkey&tim
  console.log('Example 1:', url1);


  /// Example II (Accessing an API that requires appkey, requestID and timestamp pa
  const parameter2 = {
      appkey: 'example_appkey',
      requestid: 'example_requestid',
      /// The timestamp provided by the user should be in seconds, and the gap betw
      // Example timestamp:
      // timestamp: '1717639699',
      // It is recommended to use the following statement to generate the current t
      timestamp: Math.floor(Date.now() / 1000), // Use the current timestamp (unit:
```

```
  };
  const url2 = GenReqURL(parameter2, accesstoken, wssUrl);
  // The output with the example timestamp should be as follows:
  // Example 2:wss://api.example.com/v2/ws/ivh/example_uri?appkey=example_appkey&re
  console.log('Example 2:', url2);
}


main();
```

Avatar Image Customization and Voice Clone API Documentation
Overview

Last updated：2024-07-18 17:43:40

This documentation primarily describes the open interface protocol for the **Tencent Cloud Xiaowei Avatar Customization** aPaaS platform.

Interface Calling Methods

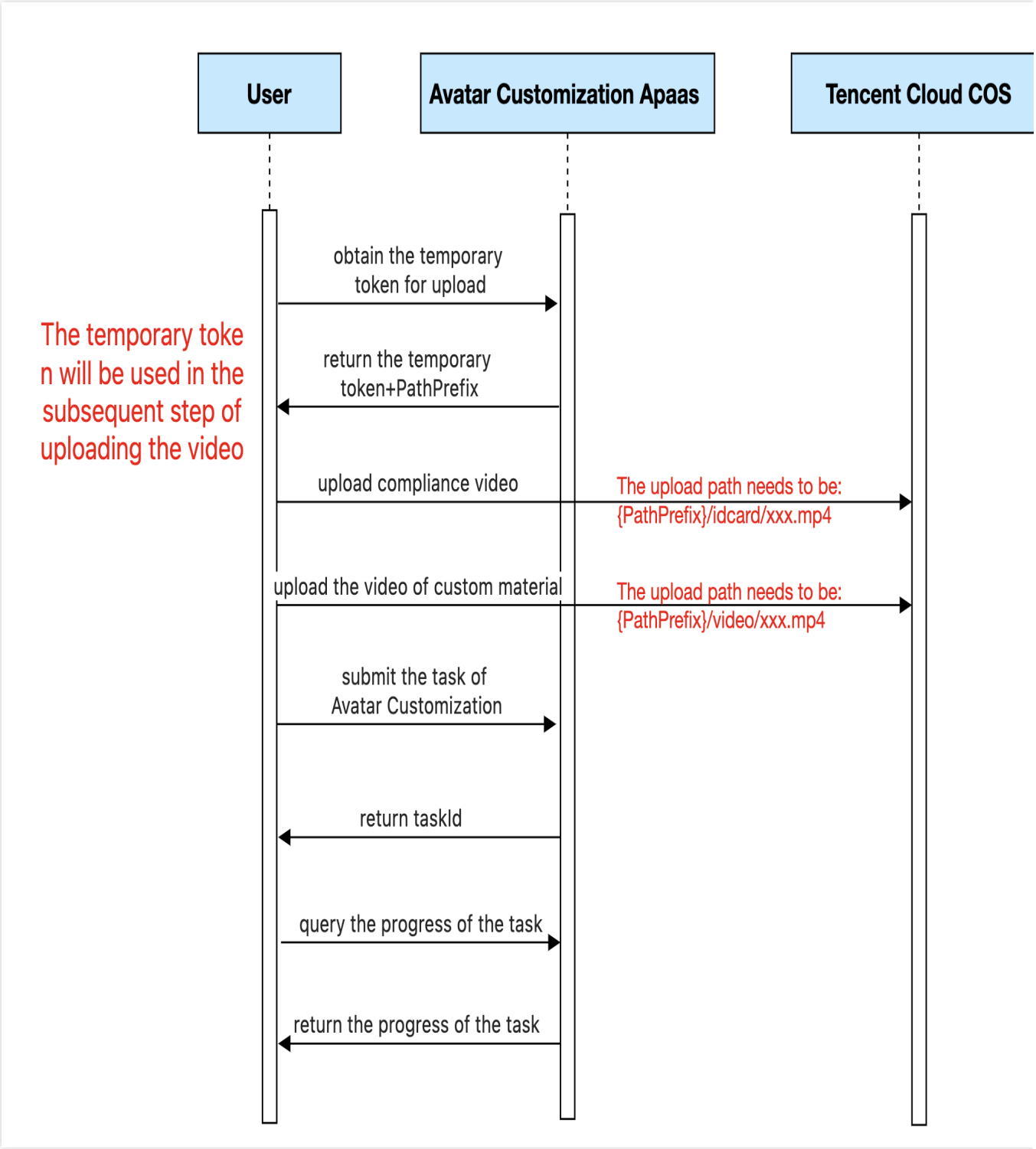
Avatar aPaas Interface Calling Method

Note:

If you purchase directly through Tencent Cloud billing, please check the appkey and token in Message Center.

Avatar Customization
API Calling Logic Diagram

Last updated：2024-07-18 17:43:59


User
Avatar Customization Apaas
Tencent Cloud COS
obtain the temporary token for upload
return the temporary token+PathPrefix
The temporary token will be used in the subsequent step of uploading the video
upload compliance video
The upload path needs to be: {PathPrefix}/idcard/xxx.mp4
upload the video of custom material
The upload path needs to be: {PathPrefix}/video/xxx.mp4
submit the task of Avatar Customization
return taskId
query the progress of the task
return the progress of the task

Obtaining the Temporary Upload Token

Last updated：2024-07-18 17:52:18

Before calling the Customized API, upload the material files to the specified cloud storage. First, obtain the temporary upload token and the specified upload path through this API, then upload the corresponding files to the cloud storage using the Upload Materials to Tencent Cloud COS API.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/customservice/getuploadcredentials
Header Content-Type: application/json;charset=utf-8

Request Parameters

No

Response Parameter

Parameters	Type	Mandatory	Description
Credentials	object	Yes	Temporary Credential Information
Credentials.Token	string	Yes	Temporary Credential Token
Credentials.TmpSecretId	string	Yes	Temporary Certificate Key ID
Credentials.TmpSecretKey	object	Yes	Temporary Certificate Key
ExpiredTime	int	Yes	The validity period of the temporary certificate, returned as a Unix timestamp in seconds.
PathPrefix	string	Yes	Prefix for the upload path on COS

Response Sample

©2013-2025 Tencent Cloud International Pte. Ltd.

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Credentials": {
            "Token": "XXX",
            "TmpSecretId": "XXX",
            "TmpSecretKey": "XXX"
        },
        "ExpiredTime": 1547696355,
        "PathPrefix": "domain name/customer-pipeline/{digit}/{uuid}/"
    }
}
```

Uploading Materials to Tencent Cloud COS

Last updated：2024-07-18 17:52:41

Integrate the Tencent Cloud COS SDK and call the putObject method to upload materials. The upload address and the upload credentials are needed for this process.

Tencent Cloud COS SDK documentation: COS-SDK Overview.

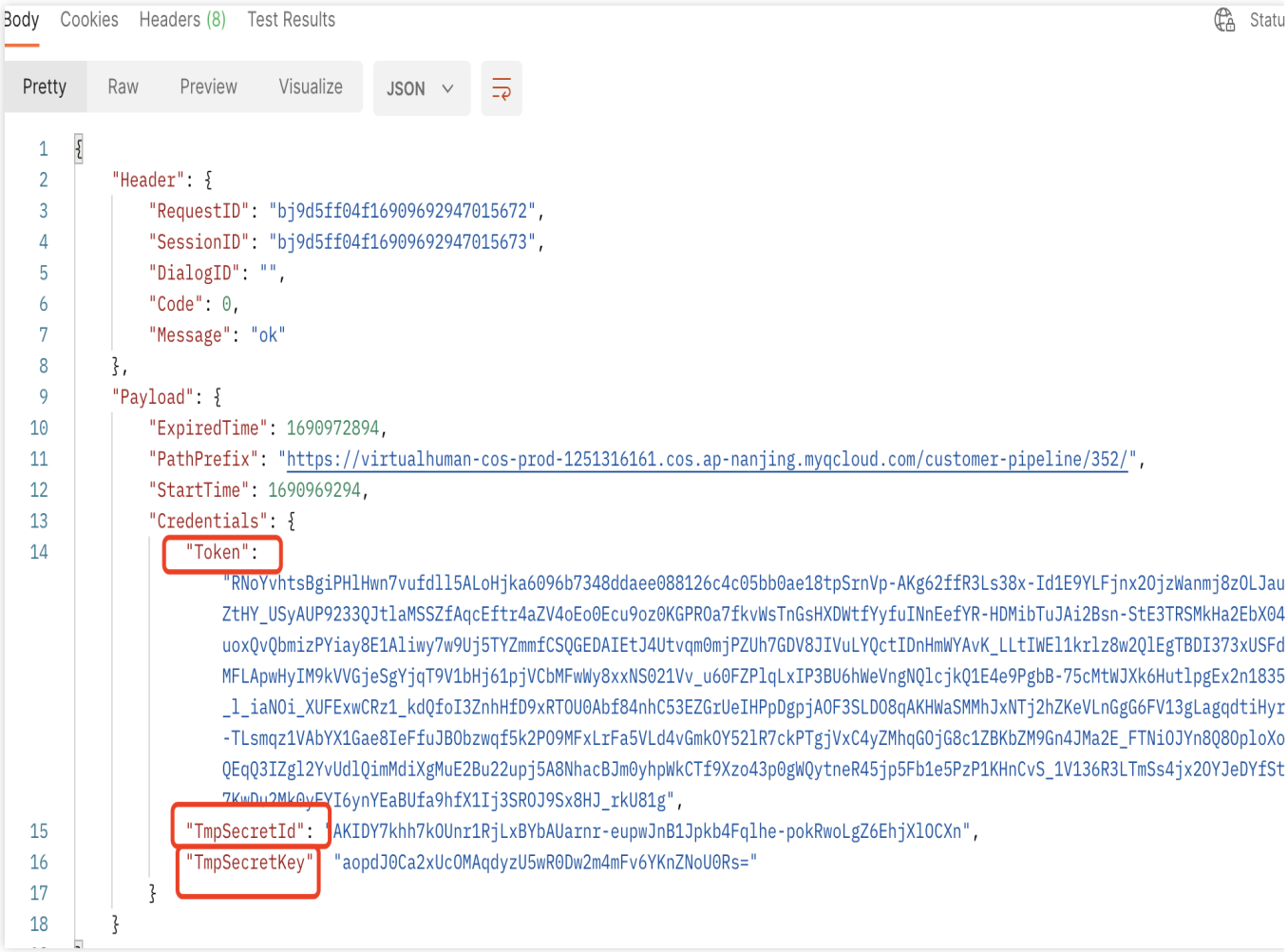
For a Java upload example, see COS - Uploading Objects.

For examples in other languages, see COS-SDK Overview.

Note:

When the upload object method in the SDK is called, temporary credentials and bucket information are needed, which can all be obtained through the Obtain the Temporary Token for Uploading API. The following examples illustrate this.

Temporary credentials information is shown in the figure below:



Bucket information is shown in the figure below:

{
"Header": {
"RequestID": "gz759824901688095927211 6935",
"SessionID": "gz759824901688095927211 6936",
"DialogID": "",
"Code": 0,
"Message": "ok"
},
"Payload": {
"PathPrefix": "https://virtualhuman-cos-prod-1251316161.cos.ap-nanjing.myqcloud.com/customer-pipeline/352/",
"Credentials": {
"TmpSecretId": "AKIDI7tPlKLJ4uJ_3KXk7q9a5p6h4XuS2hbf99_1euVt2qy-C16KJ1xDGjMw3koADSSD",
"TmpSecretKey": "JbV8fF5+6IN6TvjOdzJscCvGzRPvpcDzyU1lLB+fL+g=",
"Token":
Note: The content returned by the prefix is a part of the key, and specific file names need to be added afterwards, for example append: idcard/xxx.mp4.
bucket
region
key

Customization API

Last updated：2025-07-03 11:32:22

Use this API to submit customization requests. Query the stages of customization and related information through the Progress Query API.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/make

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
AnchorName	string	Yes	Anchor name: 1. This name is mainly used to identify the custom avatar/sound and can be customized depending on the actual situation. 2. Naming reference: If the number of customizations for the anchor is only one, it can be named directly after the anchor, such as "Zhang San"; you can also add the name of the clothing for better identification, such as "Zhang San blue suit". 3. No more than 50 characters, at least 2 characters, only allow Chinese characters, letters, numbers, underscores, and hyphens. 4. Must be unique.
MakeType	string	Yes	Customization categories: 1. IMAGE: 2D small sample - standard version (Studio Lip-Sync) avatar customization. 2. IMAGE_GENERAL: 2D small sample - standard version (general lip movement) avatar customization. 3. IMAGE_4K: 2D small sample - high-precision version (Studio Lip-Sync) avatar customization. 4. IMAGE_PHOTO: 2D small sample (photo avatar) avatar customization. 5. VOICE: Timbre Clone (Basic Version).

			6. ZERO_SHOT_VOICE: Timbre Clone (Ultra-fast Version).
IdentityCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityWrittenCosUrl or another customization type, or both. Requirements for the URL address of the video format authorization letter: 1. The URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the idcard path added, such as: domain name/customer-pipeline/{number}/{uuid}/idcard/a.mp4. 2. This format is mainly suitable for oral authorization letters, or you can submit written authorization letters as clear and complete videos.
IdentityWrittenCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityCosUrl or another customization type, or both. Requirements for the URL address of the PDF format authorization letter: 1. The URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the idcard path added, such as: domain name/customer-pipeline/{number}/{uuid}/idcard/b.pdf 2. This format is mainly suitable for "written authorization letter" and should be submitted as a clear and complete scanned copy.
MaterialCosUrl	string	No	Except for the ZERO_SHOT_VOICE customization type, all other customization types are required. **Avatar customization material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the video path added, such as /customer-pipline/{number}/{uuid}/video/c.mp4. 2. Video size, no more than 5GB, 4K no more than 10GB. 3. Video duration: Exclusive lip sync version: custom duration of 2-10 minutes General lip sync version: custom duration of 1-10 minutes High-precision version customization: duration of 2-10 minutes 4. Video resolution: 1080P, 4K (3840x2160). For high-precision version customization, it must be 4K. 5. Video aspect ratio: 16:9 (or 9:16)

			6. Video frame rate: No less than 25 fps and no more than 60 fps. 7. Video format: MP4 and MOV **Customized Text To Speech material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the audio path added, such as /customer-pipline/{number}/{uuid}/audio/c.zip. 2. Compressed Archive Format: .zip format, each zip package for customizing one voice type; do not create new folders when compressing, select all wav files directly for compression. 3. Notes for audio files within a single zip file: Audio quantity: A zip file can contain one or more wav formatted audio files, with a total amount no more than 10. Audio size: The total size of the audio files in each zip file is no more than 1G. Audio format: Each audio file must be in wav format. For other audio formats, convert to wav format before recompressing into zip. Audio sample rate: A sampling rate equal to or greater than 24k is sufficient, recommend 24K or 36K. Audio naming: Name should not contain spaces or special characters, extension must be lowercase ".wav". **Photo Digital Human customization material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the photo path added, such as /customer-pipline/{number}/{uuid}/photo/example.png. 2. Image name: no less than 2 characters, only allow Chinese characters, letters, digits, underscores, hyphens. Image format: support jpg, jpeg, png, webp. Image size: no more than 16M. Image aspect ratio: support 1:1, 9:16, 16:9, 4:3. 3. Clear front view of the person, face-centered, natural expression, mouth closed.
IsHaveBackground	bool	No	Avatar customization type: Whether the trained avatar retains the original background. The default is "No", meaning without retaining the original background, and the background can be changed as needed during application.
SexType	string	Yes	Gender:

			MALE: Male FEMALE: Female
Notes	string	No	Customized remarks, within 100 characters.
TextDriver	string	No	Text content used to generate the driving demo, with a character length between 4 and 1000 (including SSML tags, each Chinese character counts as one character).
VoiceDriverCosFile	string	No	Requirements for the audio file path for generating the driving demo: 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the audio path added, such as /customer-pipline/{number}/{uuid}/audio/example.wav. 2. Audio file size cannot exceed 10M. Format support: wav, mp3, wma, m4a, aac.
AudioId	string	No	For the ZERO_SHOT_VOICE customization type, it is required to fill in the AudioId returned after passing the Query Audio Quality Inspection Task Progress.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The created task ID. Access the Progress Query API with the TaskId to obtain the production progress and result.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "AnchorName": "Jingxuan in a green dress, sitting pose",
        "MakeType": "IMAGE",
        "IdentityCosUrl": "XXXX",
        "MaterialCosUrl": "YYYY",
        "IsRemoveBackground": true
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "TaskId": "666"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Progress Query Interface

Last updated：2024-07-18 17:53:03

Query the task's production progress and results through the TaskId.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/getprogress

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Custom API.

Response Parameter

Parameters	Type	Mandatory	Description
Status	string	Yes	Task production status: MAKING: In progress SUCCESS: Production succeeded FAIL: Production failed
FailMessage	string	No	When the Status is FAIL, the error reason is returned for troubleshooting.
StageV2	string	No	(New Version) When the Status is MAKING, the specific stage of the task in progress is returned: SUBMIT: Material submission AUDIT: Manual review PROCESS: Preprocessing (only applicable for avatar customization) TRAIN: Intelligent training CONFIRM: Effect confirmation (not applicable for photo customization) END: Completion

©2013-2025 Tencent Cloud International Pte. Ltd.

StatusV2	string	No	When StageV2 is at a certain stage, the specific status of that stage is: WAIT_PROCESS: To be processed PROCESSING: In process SUCCESS: succeeded FAIL: Failed
StageInfo	string	No	Indicates extra information for a certain stage, displayed as strings in JSON format. JSON format types for different stages are listed in the table below. Currently, only the effect confirmation stage (StageV2: CONFIRM) has values.
AssetList	Array of [AssetInfo]	No	Returns the customized image or voice information. Currently supported customization types are as follows: IMAGE_PHOTO: Photo Avatar image customization ZERO_SHOT_VOICE: Voice Clone (ultra edition)

Extra StageInfo information for different stages is shown in the table below:

StageV2	Type	Description	Example
CONFIRM: effect confirmation stage	string of [CustomerConfirmContent]	When the stage is at the effect confirmation stage (StageV2: CONFIRM) and the status is to be processed (StatusV2: WAIT_PROCESS), a demo video of the effect to be confirmed is returned. The customer can call the Effect Confirmation API to approve or reject the effect of the demo video based on the returned video.	{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"Reje[{\\"StartTime\\":\\"2023-11-01[\\"url1\\",\\"url2\\"]}]}

CustomerConfirmContent

Parameters	Type	Mandatory	Description
TrainResult	Array of [string]	Yes	URL address of the demo video for customers to confirm the effect
RejectReason	string	Yes	Reasons for the digital human side not accepting retraining
TrainRecord	Array of [TrainRecord]	No	List of demo videos of historical training

TrainRecord

Parameters	Type	Mandatory	Description
StartTime	string	Yes	Historical training time
Record	Array of [string]	Yes	URL address of demo videos of historical training

AssetInfo

Parameters	Type	Mandatory	Description
VirtualmanTypeCode	string	No	Avatar type code (see Appendix 3 - Avatar Types for details), returned by image customization
VirtualmanKey	string	No	Avatar virtualmankey, returned by image customization
Resolution	string	No	Avatar resolution, returned by image customization
OriginZoom	string	No	Scale factor, rounded to three decimal places, returned by image customization
VirtualmanResourceId	int32	No	Avatar resource ID, returned by image customization
TimbreKey	string	No	Voice key, returned by Voice Clone
TimbreSample	string	No	Voice demo link, returned by Voice Clone
MakeType	string	Yes	Customization Types: IMAGE_PHOTO: Photo Avatar image customization ZERO_SHOT_VOICE: Voice Clone (ultra edition)

©2013-2025 Tencent Cloud International Pte. Ltd.

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666
  }
}
```

Response Sample

```
{
  "Header": {
    "RequestID": "gz1a74c25f17030396653416949",
    "SessionID": "gz1a74c25f17030396653416950",
    "DialogID": "",
    "Code": 0,
    "Message": "ok"
  },
  "Payload": {
    "StageInfo": "{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"RejectReason\\":\\"
    "Status": "MAKING",
    "StatusV2": "WAIT_PROCESS",
    "FailMessage": "",
    "StageV2": "CONFIRM"
  }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Effect Confirmation API

Last updated：2024-07-18 17:54:19

When the progress query API shows that the task is in the effect confirmation stage (Status: MAKING, StageV2: CONFIRM, StatusV2: WAIT_PROCESS), perform effect confirmation on the demo video.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/customerconfirm

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Custom API
Operate	string	Yes	Operations: AGREE: Approve REJECT: Reject
Reason	string	No	Reason to reject, required when Operate: REJECT, and up to 300 characters

Response Parameter

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666,
    "Operate": "REJECT",
    "Reason": "The video character has a green outline, so the effect is not satisf
  }
```

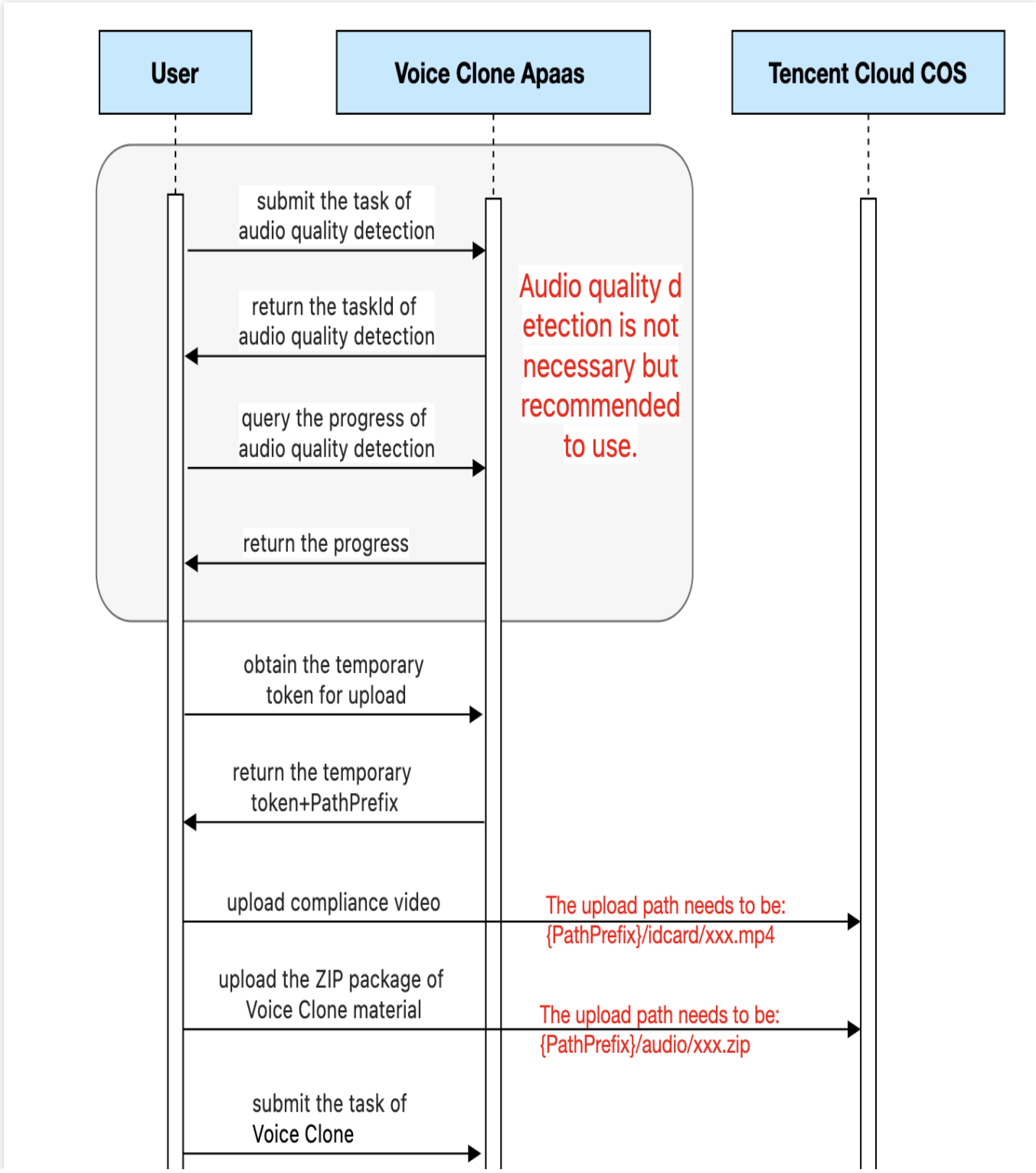
```
}
```

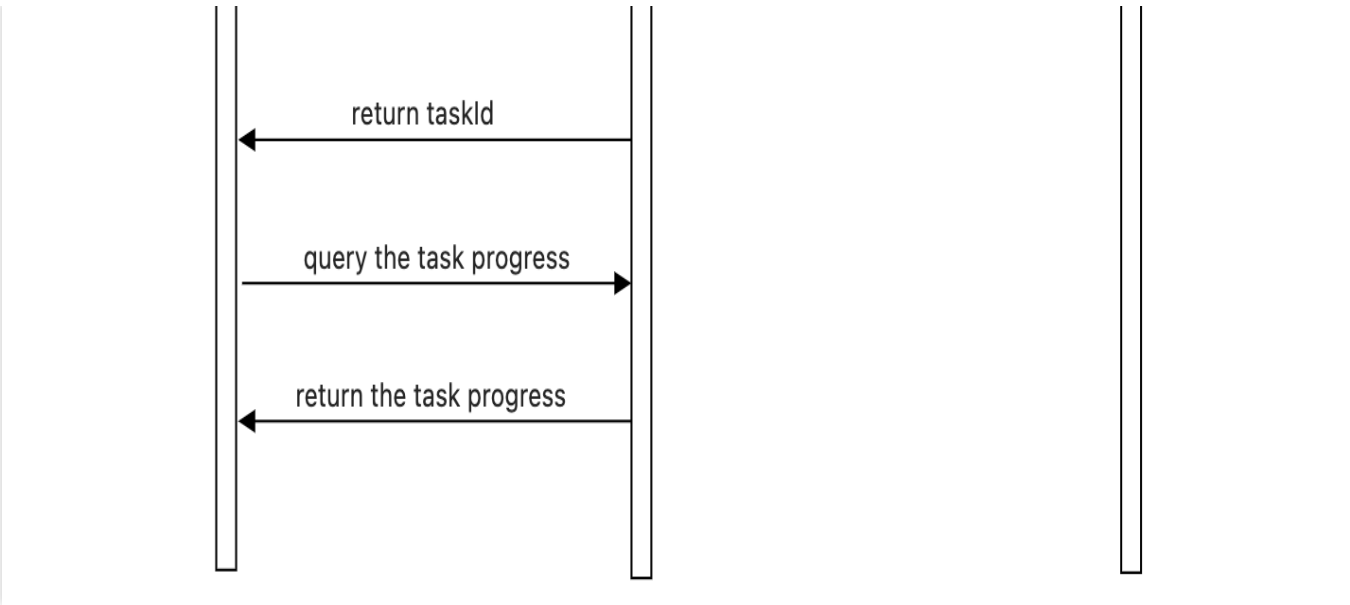
Response Sample

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "RequestID": "123"
  },
  "Payload": {
  }
}
```

Voice Clone (Basic Edition) API Calling Logic Diagram

Last updated：2024-07-18 17:54:49




return taskId
query the task progress
return the task progress

Audio Quality Inspection Task Creation API

Last updated：2024-07-18 17:55:10

This API can be used to pre-check audio quality, improving customization efficiency.
Supported audio quality metrics:
Signal-to-noise ratio: Being greater than or equal to 30 is acceptable.
Reverberation index: Being greater than or equal to 30 is acceptable.
Clipping index: Being less than or equal to 0 is acceptable.
Effective duration: Being greater than or equal to 10 minutes is acceptable.
Text accuracy: Being greater than or equal to 70% is acceptable (this metric is valid when "audio reference text" is provided).

Calling Protocol

HTTPS + JSON
POST /v2/ivh/algotaskserver/algotaskservice/audioevaluation
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
AudioUrl	string	Yes	URL of the audio to be checked. Currently, the supported audio format is WAV.
ReferenceText	string	No	Audio reference text (optional). When it is not provided, it means the "text accuracy" metric is not required.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The ID of the created audio quality inspection task

©2013-2025 Tencent Cloud International Pte. Ltd.

Request Sample

```
{
  "Header": {},
  "Payload": {
    "AudioUrl": "xxx",
    "ReferenceText": "Hello"
  }
}
```

Response Sample

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "RequestID": "123"
  },
  "Payload": {
    "TaskId": "xxx"
  }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Audio Quality Inspection Task Status Query API

Last updated：2024-07-18 17:56:25

Use the taskId to query the status and results of the audio quality inspection task.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/algotaskserver/algotaskservice/getaudioevaluationstate
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Audio Quality Inspection Task Creation API

Response Parameter

Parameters	Type	Mandatory	Description
TaskState	string	Yes	Task Status: CREATED: Created QUEUING: Queuing PROCESSING: In process SUCCESS: Task succeeded FAIL: Task failed
Result	object	Yes	Audio quality inspection task results
Result.Snr	object	Yes	Audio signal-to-noise ratio metric
Result.Snr.Value	float	Yes	Measured value of the audio signal-to-noise ratio metric

Result.Snr.QualifiedRange	Array of [float]	Yes	Acceptable range of the audio signal-to-noise ratio metric [lower_limit,upper_limit]
Result.Snr.Description	string	Yes	Description of the audio signal-to-noise ratio metric
Result.C50	object	Yes	Audio reverberation metric
Result.C50.Value	float	Yes	Measured value of the audio reverberation metric
Result.C50.QualifiedRange	Array of [float]	Yes	Acceptable range of the audio reverberation metric [lower_limit,upper_limit]
Result.C50.Description	string	Yes	Description of the audio reverberation metric
Result.Clipping	object	Yes	Audio clipping metric
Result.Clipping.Value	float	Yes	Measured value of the audio clipping metric
Result.Clipping.QualifiedRange	Array of [float]	Yes	Acceptable range of the audio clipping metric [lower_limit,upper_limit]
Result.Clipping.Description	string	Yes	Description of the audio clipping metric
Result.Duration	object	Yes	Audio duration metric
Result.Duration.Value	float	Yes	Audio duration
Result.Duration.QualifiedRange	Array of [float]	Yes	Acceptable range of the audio duration [lower_limit,upper_limit]
Result.Duration.Description	string	Yes	Description of the audio duration metric
Result.Wer	object	Yes	Audio text accuracy metric
Result.Wer.Value	float	Yes	Text accuracy, only valid when the "reference text" is not empty.
Result.Wer.QualifiedRange	Array of [float]	Yes	Acceptable range of text accuracy [lower_limit,upper_limit]
Result.Wer.Description	string	Yes	Description of text accuracy
Result.AlignDetails	array of	Yes	Text alignment result

©2013-2025 Tencent Cloud International Pte. Ltd.

	[TextAlign]		
ErrorMsg	string	Yes	Task failure prompt message

TextAlign

Parameters	Type	Mandatory	Description
Detail	Array of [string]	Yes	Text alignment result for each character, such as ["good","good"]/["good","<eps>"], and "<eps>" is a placeholder.

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": "xxx"
  }
}
```

Response Sample

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "RequestID": "123"
  },
  "Payload": {
    "Status": "SUCCESS",
    "Result": {
      "Snr": {
        "Value": 72.67048,
        "QualifiedRange": [
          30,
          2147483600
        ],
        "Description": "Audio signal-to-noise ratio; the higher, the better."
      },
      "C50": {
```

```
      "Value": 57.082405,
      "QualifiedRange": [
        30,
        2147483600
      ],
      "Description": "Reverberation metric; the higher, the better."
    },
    "Clipping": {
      "QualifiedRange": [
        -2147483600,
        0
      ],
      "Description": "Clipping metric: Being less than or equal to 0 is acceptabl
      "Value": 0
    },
    "Duration": {
      "Value": 24.21,
      "QualifiedRange": [
        600,
        2147483600
      ],
      "Description": "Effective duration: It is measured in seconds, and being gr
    },
    "Wer": {
      "Value": 0.91358024,
      "QualifiedRange": [
        0.7,
        1
      ],
      "Description": "Text accuracy: Being greater than or equal to 70% is accept
    },
    "AlignDetails": [
      {
        "Detail": [
          "<eps>",
          "Big"
        ]
      },
      {
        "Detail": [
          "Family"
          "Family"
        ]
      },
      {
        "Detail": [
          "<eps>",
```

```
            "Good"
          ]
        }
      ]
    },
    "ErrorMsg": ""
  }
}
```

Obtaining the Temporary Upload Token

Last updated：2024-07-18 17:57:37

Before calling the Customized API, upload the material files to the specified cloud storage. First, obtain the temporary upload token and the specified upload path through this API, then upload the corresponding files to the cloud storage using the Upload Materials to Tencent Cloud COS API.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/customservice/getuploadcredentials
Header Content-Type: application/json;charset=utf-8

Request Parameters

No

Response Parameter

Parameters	Type	Mandatory	Description
Credentials	object	Yes	Temporary Credential Information
Credentials.Token	string	Yes	Temporary Credential Token
Credentials.TmpSecretId	string	Yes	Temporary Certificate Key ID
Credentials.TmpSecretKey	object	Yes	Temporary Certificate Key
ExpiredTime	int	Yes	The validity period of the temporary certificate, returned as a Unix timestamp in seconds.
PathPrefix	string	Yes	Prefix for the upload path on COS

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Credentials": {
            "Token": "XXX",
            "TmpSecretId": "XXX",
            "TmpSecretKey": "XXX"
        },
        "ExpiredTime": 1547696355,
        "PathPrefix": "domain name/customer-pipeline/{digit}/{uuid}/"
    }
}
```

Uploading Materials to Tencent Cloud COS.

Last updated：2024-07-18 17:57:53

Connect the Tencent Cloud COS SDK and call the putObject method to upload materials. The upload address and the upload credentials are needed for this process.

Tencent Cloud COS SDK documentation: COS-SDK Overview.

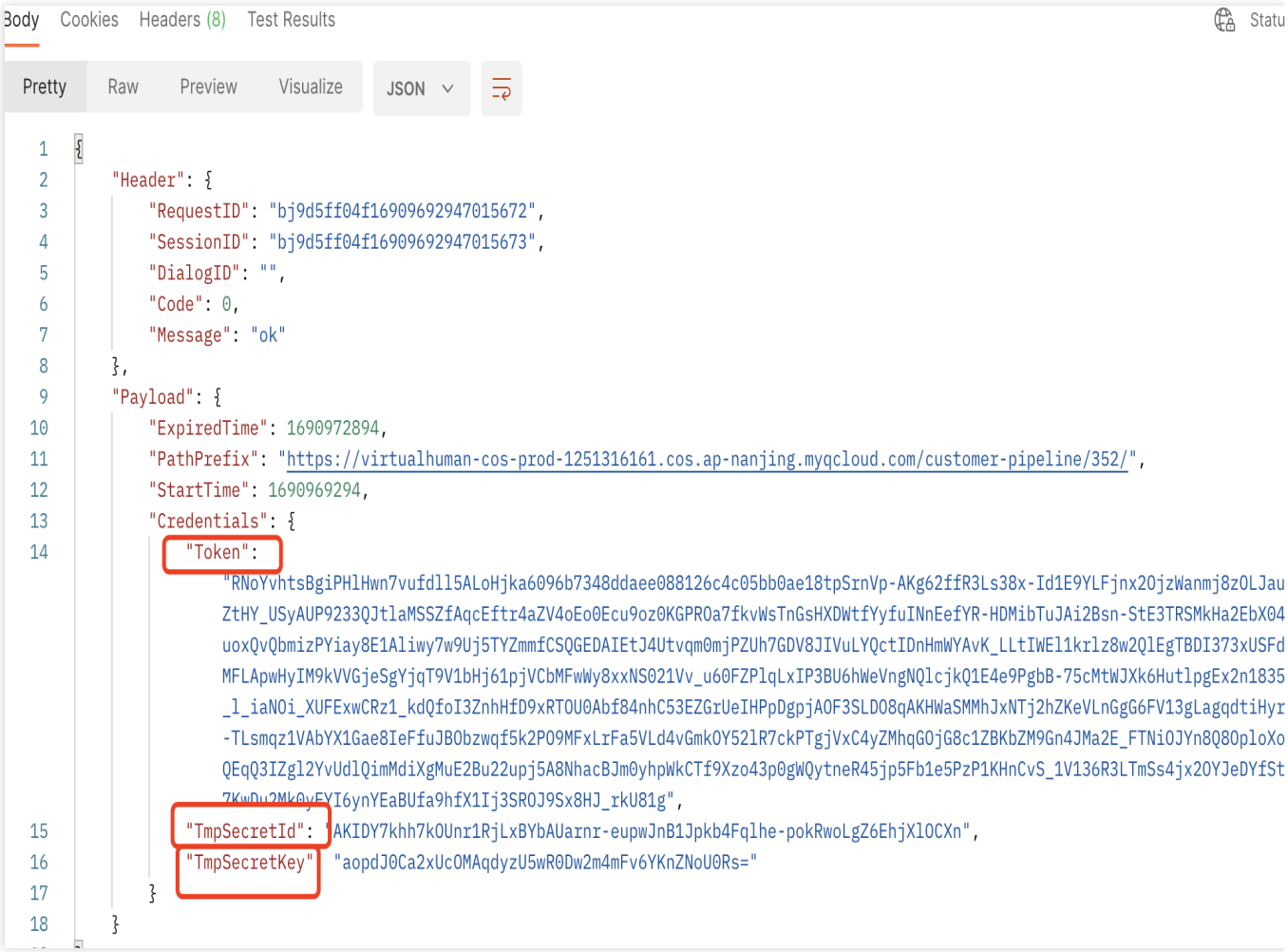
For details about Java upload examples, see COS - Uploading Objects.

For details about the examples of other languages, see COS-SDK Overview.

Note:

When the upload object method in the SDK is called, temporary credentials and bucket information are needed, which can all be obtained through the Obtain the Temporary Token for Uploading API. The following examples illustrate this.

Temporary credentials information is shown in the figure below:



Bucket information is shown in the figure below:

{
"Header": {
"RequestID": "gz759824901688095927211 6935",
"SessionID": "gz759824901688095927211 6936",
"DialogID": "",
"Code": 0,
"Message": "ok"
},
"Payload": {
"PathPrefix": "https://virtualhuman-cos-prod-1251316161.cos.ap-nanjing.myqcloud.com/customer-pipeline/352/",
"Credentials": {
"TmpSecretId": "AKIDI7tPlKLJ4uJ_3KXk7q9a5p6h4XuS2hbf99_1euVt2qy-C16KJ1xDGjMw3koADSSD",
"TmpSecretKey": "JbV8fF5+6IN6TvjOdzJscCvGzRPvpcDzyU1lLB+fL+g=",
"Token":
Note: The content returned by the prefix is a part of the key, and specific file names need to be added afterwards, for example append: idcard/xxx.mp4.
bucket
region
key

Customization API

Last updated：2025-07-03 11:32:52

Use this API to submit customization requests. Query the stages of customization and related information through the Progress Query API.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/make

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
AnchorName	string	Yes	Anchor name: 1. This name is mainly used to identify the custom avatar/sound and can be customized depending on the actual situation. 2. Naming reference: If the number of customizations for the anchor is only 1, it can be named directly after the anchor, such as "Zhang San", or add the name of the clothing for better identification, such as "Zhang San blue suit". 3. No more than 50 characters, no less than 2 characters, only allow Chinese characters, letters, digits, underscores, and hyphens. 4. Must be unique.
MakeType	string	Yes	Customization categories IMAGE: 2D small sample - standard version (Studio Lip-Sync) avatar customization. IMAGE_GENERAL: 2D small sample - standard version (Instant Lip-Sync) avatar customization. IMAGE_4K: 2D small sample - high-precision version (Studio Lip-Sync) avatar customization. IMAGE_PHOTO: 2D small sample (photo avatar) avatar customization. VOICE: voice clone (basic version). ZERO_SHOT_VOICE: voice clone (ultra-fast version).

©2013-2025 Tencent Cloud International Pte. Ltd.

IdentityCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityWrittenCosUrl or another customization type, or both. Requirements for the URL address of the video format authorization letter: 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the idcard path added, for example: domain name/customer-pipline/{number}/{uuid}/idcard/a.mp4. 2. This format is mainly suitable for "oral authorization letter" and can also submit "written authorization letter" as clear and complete videos.
IdentityWrittenCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityCosUrl or another customization type, or both. Requirements for the URL address of the PDF format authorization letter: 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the idcard path added, for example: domain name/customer-pipline/{number}/{uuid}/idcard/b.pdf. 2. This format is mainly suitable for "written authorization letter" and can be submitted as a clear and complete scanned copy.
MaterialCosUrl	string	No	Except for the ZERO_SHOT_VOICE customization type, all other customization types are required. **Avatar customization material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the video path added, for example: /customer-pipline/{number}/{uuid}/video/c.mp4. 2. Video size, no more than 5GB, 4K not more than 10GB. 3. Video duration: exclusive lip sync version customization duration is 2-10 minutes; general lip sync version customization duration is 1-10 minutes; high-precision version customization duration is 2-10 minutes. 4. Video resolution: 1080P, 4K (3840*2160). For high-precision version customization, it must be 4K. 5. Video aspect ratio: 16:9 (or 9:16) 6. Video frame rate: No less than 25 fps and no more than 60 fps.

			7. Video format: MP4 and MOV **Customized Text To Speech material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the audio path added, for example: /customer-pipline/{number}/{uuid}/audio/c.zip. 2. Compressed file format: .zip format, one zip package for customizing one voice type; do not create new folders when compressing, select all wav files directly for compression. 3. Notes for audio files within a single zip file: Audio quantity: A zip file can contain one or more wav formatted audio files, with a total amount of no more than 10. Audio size: The total size of the audio files in each zip file is no more than 1G. Audio format: Each audio file must be in wav format. For other audio formats, convert to wav format before recompressing into zip. Audio sample rate: A sampling rate equal to or greater than 24K is sufficient, recommend 24K or 36K. Audio naming: Name should not contain spaces or special characters, extension must be lowercase ".wav". **Photo Digital Human customization material URL address requirements:** 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the photo path added, for example: /customer-pipline/{number}/{uuid}/photo/example.png. 2. Image name: no less than 2 characters, only allowed to have Chinese characters, letters, numbers, underline, horizontal line; image format: jpg, jpeg, png, webp; image size: no more than 16M; image aspect ratio: 1:1 9:16 16:9 4:3. 3. Clear front view of the person, face-centered, natural expression, mouth closed.
IsHaveBackground	bool	No	Avatar customization type: Whether the trained avatar retains the original background. The default is "No", meaning without retaining the original background, and the background can be changed as needed during application.
SexType	string	Yes	Gender: MALE: Male FEMALE: Female

Notes	string	No	Customized remarks, within 100 characters.
TextDriver	string	No	Text content used to generate the driving demo, with a character length between 4 and 1000 (including SSML tags, each Chinese character counts as one character).
VoiceDriverCosFile	string	No	Requirements for the audio file path for generating the driving demo: 1. The URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the audio path added, for example: /customer-pipline/{number}/{uuid}/audio/example.wav. 2. Audio file size cannot exceed 10M, supported formats are wav, mp3, wma, M4a, aac.
AudioId	string	No	For the ZERO_SHOT_VOICE customization type, it is required to fill in the AudioId returned after passing the Query Audio Quality Inspection Task Progress.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The created task ID. Access the Progress Query API with the TaskId to obtain the production progress and result.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "AnchorName": "Jingxuan in a green dress, sitting pose",
        "MakeType": "IMAGE",
        "IdentityCosUrl": "XXXX",
        "MaterialCosUrl": "YYYY",
        "IsRemoveBackground": true
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "TaskId": "666"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Progress Query API

Last updated：2024-07-18 18:01:03

Query the task's production progress and results through the taskId.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/getprogress

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Custom API

Response Parameter

Parameters	Type	Mandatory	Description
Status	string	Yes	Task production status: MAKING: In progress SUCCESS: Production succeeded FAIL: Production failed
FailMessage	string	No	When the Status is FAIL, the error reason is returned for troubleshooting.
StageV2	string	No	(New Version) When the Status is MAKING, the specific stage of the task in progress is returned: SUBMIT: Material submission AUDIT: Manual review PROCESS: Preprocessing (only applicable for avatar customization) TRAIN: Intelligent training CONFIRM: Effect confirmation (not applicable for photo customization) END: Completion

©2013-2025 Tencent Cloud International Pte. Ltd.

StatusV2	string	No	When StageV2 is at a certain stage, the specific status of that stage is: WAIT_PROCESS: To be processed PROCESSING: In process SUCCESS: Succeeded FAIL: Failed
StageInfo	string	No	Indicates extra information for a certain stage, displayed as strings in JSON forma. JSON format types for different stages are listed in the table below. Currently, only the effect confirmation stage (StageV2: CONFIRM) has values.
AssetList	Array of [AssetInfo]	No	Returns the customized image or voice information. Currently, supported customization types are as follows: IMAGE_PHOTO: Photo Avatar image customization ZERO_SHOT_VOICE: Voice Clone (ultra edition)

Extra StageInfo information for different stages is shown in the table below:

StageV2	Type	Description	Example
CONFIRM: effect confirmation stage	string of [CustomerConfirmContent]	When the stage is at the effect confirmation stage (StageV2: CONFIRM) and the status is to be processed (StatusV2: WAIT_PROCESS), a demo video of the effect to be confirmed is returned. The customer can call the Effect Confirmation API to approve or reject the effect of the demo video based on the returned video.	{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"Reje[{\\"StartTime\\":\\"2023-11-01[\\"url1\\",\\"url2\\"]}]}

CustomerConfirmContent

Parameters	Type	Mandatory	Description
TrainResult	Array of [string]	Yes	URL address of the demo video for customer to confirm the effect
RejectReason	string	Yes	Reasons for the digital human side not accepting retraining
TrainRecord	Array of [TrainRecord]	No	List of demo videos of historical training

TrainRecord

Parameters	Type	Mandatory	Description
StartTime	string	Yes	Historical training time
Record	Array of [string]	Yes	URL address of demo videos of historical training

AssetInfo

Parameters	Type	Mandatory	Description
VirtualmanTypeCode	string	No	Avatar type code (see Appendix 3 - Avatar Types for details), returned by image customization
VirtualmanKey	string	No	Avatar virtualmankey, returned by image customization
Resolution	string	No	Avatar resolution, returned by image customization
OriginZoom	string	No	Scale factor, rounded to three decimal places, returned by image customization
VirtualmanResourceId	int32	No	Avatar resource ID, returned by image customization
TimbreKey	string	No	Voice key, returned by Voice Clone
TimbreSample	string	No	Voice demo link, returned by Voice Clone
MakeType	string	Yes	Customization Types: IMAGE_PHOTO: Photo Avatar image customization ZERO_SHOT_VOICE: Voice Clone (ultra edition)

©2013-2025 Tencent Cloud International Pte. Ltd.

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666
  }
}
```

Response Sample

```
{
  "Header": {
    "RequestID": "gz1a74c25f17030396653416949",
    "SessionID": "gz1a74c25f17030396653416950",
    "DialogID": "",
    "Code": 0,
    "Message": "ok"
  },
  "Payload": {
    "StageInfo": "{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"RejectReason\\":\\"
    "Status": "MAKING",
    "StatusV2": "WAIT_PROCESS",
    "FailMessage": "",
    "StageV2": "CONFIRM"
  }
}
```

Effect Confirmation API

Last updated：2024-07-18 18:11:09

When the progress query API shows that the task is in the effect confirmation stage (Status: MAKING, StageV2: CONFIRM, StatusV2: WAIT_PROCESS), effect confirmation should be performed on the demo video.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/customerconfirm

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Custom API
Operate	string	Yes	Operations: AGREE: Approve REJECT: Reject
Reason	string	No	Reason to reject, required when Operate: REJECT, and up to 300 characters

Response Parameter

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666,
    "Operate": "REJECT",
    "Reason": "The video character has a green outline, so the effect is not satisf
  }
```

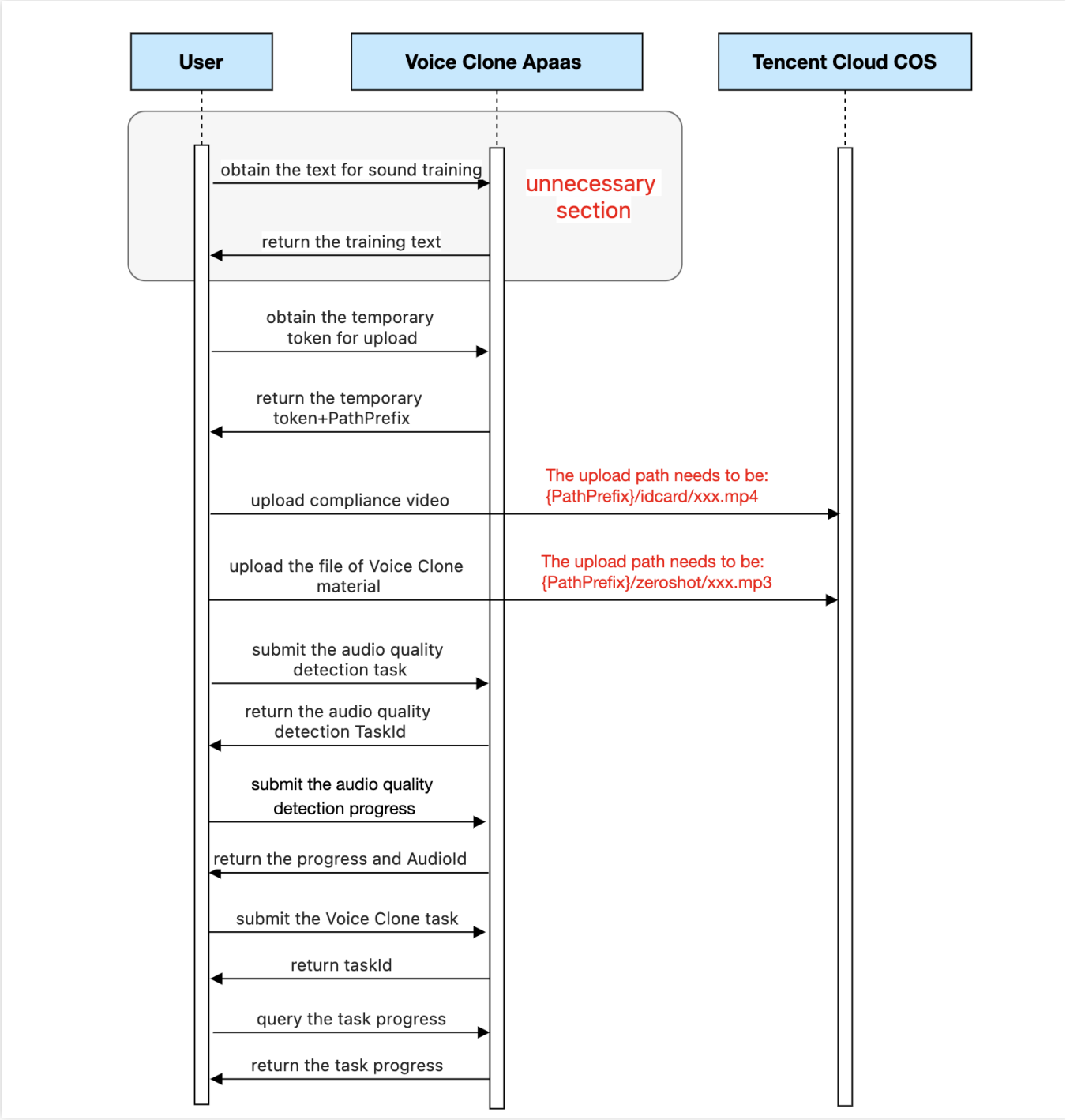
```
}
```

Response Sample

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "RequestID": "123"
  },
  "Payload": {
  }
}
```

Voice Clone (Ultra Edition) API Calling Logic Diagram

Last updated：2024-07-18 18:01:25



Obtaining the Voice Training Text

Last updated：2024-07-18 18:02:52

API Description

Call this API to obtain the document content that needs to be read aloud. Then, read this text to create an audio file, which will be used as the voice production material.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/zeroshotservice/gettext
Header Content-Type: application/json;charset=utf-8

Request Parameters

No

Response Parameter

Parameters	Type	Mandatory	Description
Texts	Array of [Text]	Yes	Selectable Text List

Text

Parameters	Type	Mandatory	Description
TextId	string	Yes	Text ID
Text	string	Yes	Text

Response Sample

©2013-2025 Tencent Cloud International Pte. Ltd.

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Texts": [
            {
                "TextId": "ec28499a-cf6d-4f7b-b7fa-4a5a54662b38",
                "Text": "Among the more than 300 products with discounts, meat snac
            }
        ]
    }
}
```

Obtaining the Temporary Upload Token

Last updated：2024-07-18 18:03:50

Before calling the Customized API, upload the material files to the specified cloud storage. First, obtain the temporary upload token and the specified upload path through this API, then upload the corresponding files to the cloud storage using the Upload Materials to Tencent Cloud COS API.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/customservice/getuploadcredentials
Header Content-Type: application/json;charset=utf-8

Request Parameters

No

Response Parameter

Parameters	Type	Mandatory	Description
Credentials	object	Yes	Temporary Credential Information
Credentials.Token	string	Yes	Temporary Credential Token
Credentials.TmpSecretId	string	Yes	Temporary Certificate Key ID
Credentials.TmpSecretKey	object	Yes	Temporary Certificate Key
ExpiredTime	int	Yes	The validity period of the temporary certificate, returned as a Unix timestamp in seconds.
PathPrefix	string	Yes	Prefix for the upload path on COS

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Credentials": {
            "Token": "XXX",
            "TmpSecretId": "XXX",
            "TmpSecretKey": "XXX"
        },
        "ExpiredTime": 1547696355,
        "PathPrefix": "domain name/customer-pipeline/{digit}/{uuid}/"
    }
}
```

Uploading Materials to Tencent Cloud COS

Last updated：2024-07-18 18:04:14

Connect the Tencent Cloud COS SDK and call the putObject method to upload materials. The upload address and the upload credentials are needed for this process.

Tencent Cloud COS SDK documentation: COS-SDK Overview.

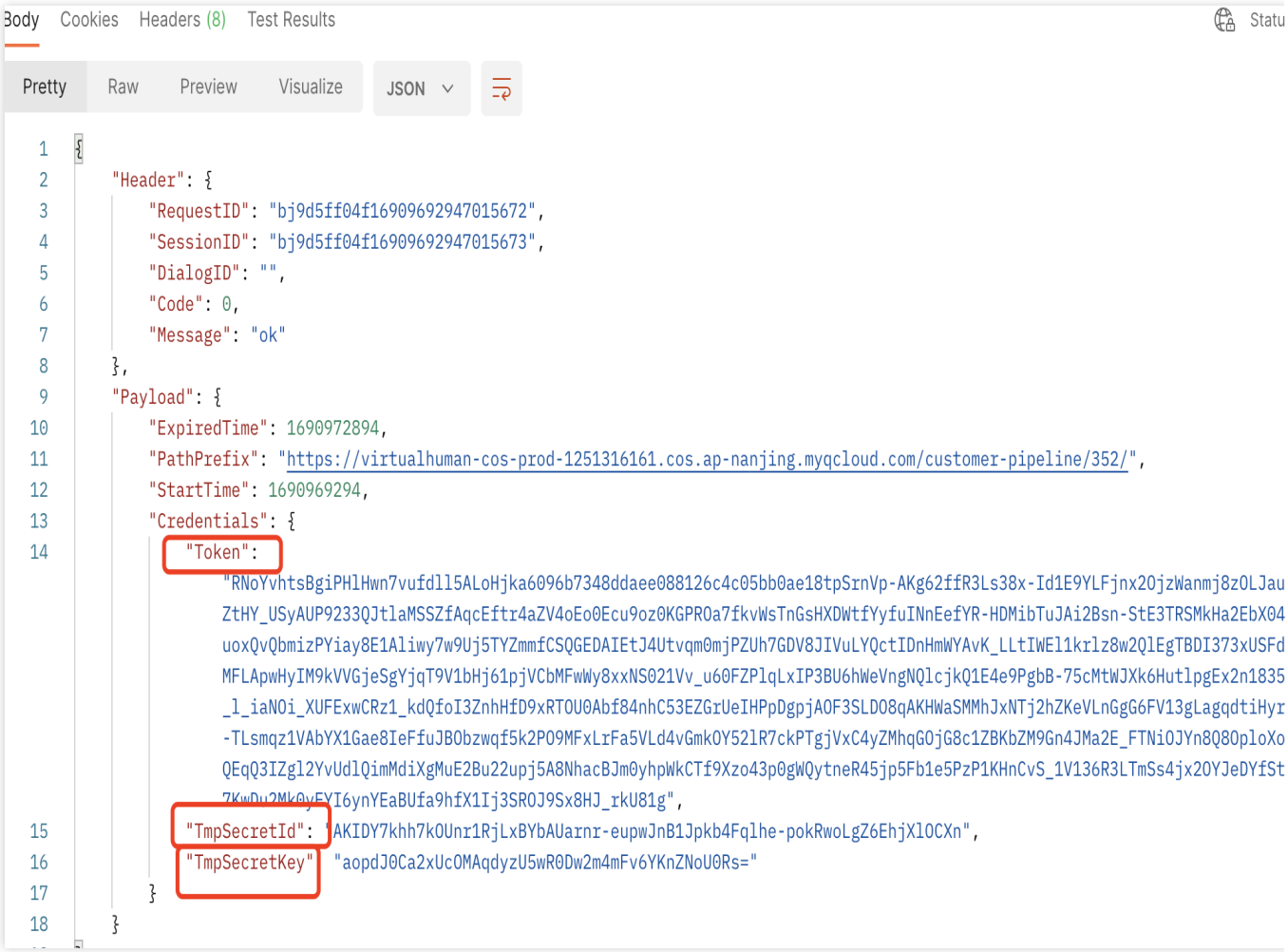
For details about Java upload examples, see COS - Uploading Objects.

For details about examples of other languages, see COS-SDK Overview.

Note:

When the upload object method in the SDK is called, temporary credentials and bucket information are needed, which can all be obtained through the Obtain the Temporary Token for Uploading API. The following examples illustrate this.

Temporary credentials information is shown in the figure below:



Bucket information is shown in the figure below:

{
"Header": {
"RequestID": "gz759824901688095927211 6935",
"SessionID": "gz759824901688095927211 6936",
"DialogID": "",
"Code": 0,
"Message": "ok"
},
"Payload": {
"PathPrefix": "https://virtualhuman-cos-prod-1251316161.cos.ap-nanjing.myqcloud.com/customer-pipeline/352/",
"Credentials": {
"TmpSecretId": "AKIDI7tPlKLJ4uJ_3KXk7q9a5p6h4XuS2hbf99_1euVt2qy-C16KJ1xDGjMw3koADSSD",
"TmpSecretKey": "JbV8fF5+6IN6TvjOdzJscCvGzRPvpcDzyU1lLB+fL+g=",
"Token":
Note: The content returned by the prefix is a part of the key, and specific file names need to be added afterwards, for example append: idcard/xxx.mp4.
bucket
region
key

Submitting Audio Quality Inspection Task

Last updated：2024-07-18 18:05:56

Read aloud the content from Obtain Voice Training Text, record the audio file, and upload it to the specified COS path. After the upload, the audio must undergo a quality inspection. The audio that passes the quality inspection will return an AudioId which is used to call the customization API for the customization of the voice speaking one sentence.

Invocation Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/zeroshotservice/detectaudioquality
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TextId	string	No	TextId returned by the Obtain Voice Training Text API.
ReferenceText	string	No	Audio reference text. ReferenceText and TextId are two different ways to obtain voice training text. You can leave both fields blank or choose one of the two options for more accurate audio quality inspection results.
AudioUrl	string	Yes	Voice file URL requirements: 1: The URL address should be the resource URL uploaded to the specified path through 4.1, with an added zeroshot path, such as /customer-pipeline/{digit}/{uuid}/zeroshot/xxx.mp3. 2: The recommended audio length for upload is 10-30 seconds, and a single file should not exceed 20 MB. 3: Supported audio formats are wav, mp3, aac, m4a, wma, and asf. The sampling rate should be greater than 16 K. For compressed formats, a bitrate higher than 128 kbps is recommended. 4: The audio file name should be 2-50 characters long, and can only contain Chinese characters, letters, numbers, underscores, and hyphens.

©2013-2025 Tencent Cloud International Pte. Ltd.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The ID of the created task. Use the taskId to access the Query Audio Quality Inspection Task Progress API to obtain the progress and results of the quality inspection.

Request Sample

```
{
  "Header": {},
  "Payload": {
    "AudioId": "XXXX",
    "AudioUrl": "YYYY"
  }
}
```

Response Sample

```
{
  "Header": {
    "Code": 0,
    "DialogID": "",
    "Message": "",
    "RequestID": "123"
  },
  "Payload": {
    "TaskId": "666"
  }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Querying Audio Quality Inspection Task Progress

Last updated：2024-07-18 18:07:16

Use the taskId to query the progress and results of the audio quality inspection.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/assetmanager/zeroshotservice/getdetectprogress
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by submitting the audio quality inspection task

Response Parameter

Parameters	Type	Mandatory	Description
AudioId	string	Yes	Audio ID
DetectionCode	int	Yes	Detection codes: 0: indicating that the current audio passed. -1: indicating that detection failed and needs to be retried. -2: indicating that audio did not pass; user is prompted to re-record (usually due to missing, incorrect, or extra reading). -3: indicating that significant noise in the audio, did not pass.
DetectionMsg	string	No	Detection prompt message

©2013-2025 Tencent Cloud International Pte. Ltd.

TaskState	string	No	Detection task status: PROCESSING: In progress SUCCESS: Task processing completed FAIL: Task processing failed

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666
  }
}
```

Response Sample

```
{
  "Header": {
    "RequestID": "gz1a74c25f17030396653416949",
    "SessionID": "gz1a74c25f17030396653416950",
    "DialogID": "",
    "Code": 0,
    "Message": "ok"
  },
  "Payload": {
    "TaskState": "SUCCESS",
    "AudioId": "ZZZ",
    "DetectionCode": 0,
    "DetectionMsg":"success"
  }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Customization API

Last updated：2025-07-03 11:34:06

Use this API to submit customization requests. Query the stages of customization and relevant information through the Progress Query API.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/make

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
AnchorName	string	Yes	Anchor name: 1. This name is mainly used to identify the custom avatar/sound and can be customized depending on the actual situation. 2. Naming reference: If the number of customizations for the anchor is only 1, it can be named directly after the anchor, such as "Zhang San", or add the name of the clothing for better identification, such as "Zhang San blue suit". 3. No more than 50 characters, at least 2 characters. Only allow Chinese characters, letters, digits, underscores, and hyphens. 4. Must be unique.
MakeType	string	Yes	Customization categories: 1. IMAGE: 2D small sample - standard version (Studio Lip-Sync) avatar customization. 2. IMAGE_GENERAL: 2D small sample - standard version (general lip movement) avatar customization. 3. IMAGE_4K: 2D small sample - high-precision version (Studio Lip-Sync) avatar customization. 4. IMAGE_PHOTO: 2D small sample (photo avatar) avatar customization. 5. VOICE: Timbre Clone (Basic Version). 6. ZERO_SHOT_VOICE: Timbre Clone (Ultra-fast Version).

©2013-2025 Tencent Cloud International Pte. Ltd.

IdentityCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityWrittenCosUrl or another customization type, or both. Requirements for the URL address of the video format authorization letter: 1. The URL address is the resource URL uploaded to the specified path via uploading material to Tencent Cloud COS, with the idcard path added. For example, domain name/customer-pipeline/{number}/{uuid}/idcard/a.mp4. 2. This format is mainly suitable for oral authorization letters, or you can submit written authorization letters as clear and complete videos.
IdentityWrittenCosUrl	string	No	Except for the IMAGE_PHOTO and ZERO_SHOT_VOICE customization types, fill in either the IdentityCosUrl or another customization type, or both. Requirements for the URL address of the PDF format authorization letter: 1. URL address is the resource URL uploaded to the specified path by uploading material to Tencent Cloud COS, with the idcard path added. For example, domain name /customer-pipline/{number}/{uuid}/idcard/b.pdf. 2. This format is mainly suitable for "written authorization letter" and should be submitted as a clear and complete scanned copy.
MaterialCosUrl	string	No	Except for the ZERO_SHOT_VOICE customization type, all other customization types are required. **Avatar customization material URL address requirements:** 1. URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the video path added, such as /customer-pipline/{number}/{uuid}/video/c.mp4. 2. Video size, no more than 5GB, 4K not more than 10GB. 3. Video duration: exclusive lip sync version customization duration is 2-10 minutes; customizable duration for general lip sync version is 1-10 minutes; customization duration for high-precision version is 2-10 minutes. 4. Video resolution: 1080P or 4K (3840*2160); for high-precision version customization, it must be 4K. 5. Video aspect ratio: 16:9 (or 9:16) 6. Video frame rate: No less than 25 fps and no more than 60 fps.

			7. Video format: MP4 and MOV **Customized Text To Speech material URL address requirements:** 1. URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the audio path added, such as /customer-pipline/{number}/{uuid}/audio/c.zip. 2. Compressed file format: zip format, one zip package for customizing one voice type; do not create new folders when compressing, select all wav files directly for compression. 3. Notes for audio files within a single zip file: Audio quantity: A zip file can contain one or more wav formatted audio files, with a total amount no more than 10. Audio size: The total size of the audio files in each zip file is no more than 1G. Audio format: Each audio file must be in wav format. For other audio formats, convert to wav format before recompressing into zip. Audio sample rate: The sampling rate should be equal to or greater than 24K, recommend 24K or 36K. Audio naming: Name should not contain spaces or special characters, extension should be lowercase ".wav". **Photo Digital Human customization material URL address requirements:** 1. URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the photo path added, such as /customer-pipline/{number}/{uuid}/photo/example.png. 2. Image name: at least 2 characters, only allow Chinese characters, letters, digits, underscores, hyphens. Image format: support jpg, jpeg, png, webp. Image size: no more than 16M. Image aspect ratio: support 1:1, 9:16, 16:9, 4:3. 3. Clear front view of the person, face-centered, natural expression, mouth closed.
IsHaveBackground	bool	No	Image customization type: Whether the trained image retains the original background. The default is "No", meaning that the original background is not retained, and the background can be changed as needed during application.
SexType	string	Yes	Gender: MALE: Male FEMALE: Female

Notes	string	No	Customized remarks, within 100 characters.
TextDriver	string	No	Text content used to generate the driving demo, with a character length between 4 and 1000 (including SSML tags, each Chinese character counts as one character).
VoiceDriverCosFile	string	No	Requirements for the audio file path for generating the driving demo: 1. URL address is the resource URL uploaded to the specified path via upload material to Tencent Cloud COS, with the audio path added, such as /customer-pipline/{number}/{uuid}/audio/example.wav. 2. Audio file size cannot exceed 10M, format support wav, mp3, wma, m4a, aac.
AudioId	string	No	For the ZERO_SHOT_VOICE customization type, it is required to fill in the AudioId returned after passing the Query Audio Quality Inspection Task Progress.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The created task ID. Access the Progress Query API with the TaskId to obtain the production progress and result.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "AnchorName": "Jingxuan in a green dress, sitting pose",
        "MakeType": "IMAGE",
        "IdentityCosUrl": "XXXX",
        "MaterialCosUrl": "YYYY",
        "IsRemoveBackground": true
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "TaskId": "666"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Progress Query API

Last updated：2024-07-18 18:11:26

Query the task's production progress and results through the taskId.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/assetmanager/customservice/getprogress

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The taskId returned by the Custom API

Response Parameter

Parameters	Type	Mandatory	Description
Status	string	Yes	Task production status: MAKING: In process SUCCESS: Production succeeded FAIL: Production failed
FailMessage	string	No	When the Status is FAIL, the error reason is returned for troubleshooting.
StageV2	string	No	(New Version) When the Status is MAKING, the specific stage of the task in progress is returned: SUBMIT: Material submission AUDIT: Manual review PROCESS: Preprocessing (only applicable for avatar customization) TRAIN: Intelligent training CONFIRM: Effect confirmation (not applicable for photo customization) END: Completion

©2013-2025 Tencent Cloud International Pte. Ltd.

StatusV2	string	No	When StageV2 is at a certain stage, the specific status of that stage is: WAIT_PROCESS: To be processed PROCESSING: In process SUCCESS: Succeeded FAIL: Failed
StageInfo	string	No	Extra information for a certain stage, strings in JSON format, and the JSON format types for different stages are listed in the table below. Currently, only the effect confirmation stage (StageV2: CONFIRM) has values.
AssetList	Array of [AssetInfo]	No	Returns the avatar or voice information for customization. Currently supported customization types are: IMAGE_PHOTO: Photo Avatar Image Customization ZERO_SHOT_VOICE: Voice Clone (ultra edition)

Extra StageInfo information for different stages is shown in the table below:

StageV2	Type	Description	Example
CONFIRM: effect confirmation stage	string of [CustomerConfirmContent]	When the stage is at the effect confirmation stage (StageV2: CONFIRM) and the status is to be processed (StatusV2: WAIT_PROCESS), a demo video of the effect to be confirmed is returned. The customer can call the Effect Confirmation API to approve or reject the effect of the demo video based on the returned video.	{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"Reje[{\\"StartTime\\":\\"2023-11-01[\\"url1\\",\\"url2\\"]}]}

CustomerConfirmContent

Parameters	Type	Mandatory	Description
TrainResult	Array of [string]	Yes	URL address of the demo video for customer to confirm the effect
RejectReason	string	Yes	Reasons for the digital human side not accepting retraining
TrainRecord	Array of [TrainRecord]	No	List of demo videos of historical training

TrainRecord

Parameters	Type	Mandatory	Description
StartTime	string	Yes	Historical training time
Record	Array of [string]	Yes	URL address of demo videos of historical training

AssetInfo

Parameters	Type	Mandatory	Description
VirtualmanTypeCode	string	No	Digital human type code, see Appendix 3 - Types of Intelligent Humans, returned by avatar customization
VirtualmanKey	string	No	Digital human virtualmankey, returned by avatar customization
Resolution	string	No	Digital human resolution, returned by avatar customization
OriginZoom	string	No	Scale factor, rounded to three decimal places, returned by avatar customization
VirtualmanResourceId	int32	No	Digital human resource ID, returned by avatar customization
TimbreKey	string	No	Timbre key, returned by voice clone
TimbreSample	string	No	Timbre demo link, returned by voice clone
MakeType	string	Yes	Customization Types: IMAGE_PHOTO: Photo Avatar Image Customization

			ZERO_SHOT_VOICE: Voice Clone (ultra edition)

Request Sample

```
{
  "Header": {},
  "Payload": {
    "TaskId": 666
  }
}
```

Response Sample

```
{
  "Header": {
    "RequestID": "gz1a74c25f17030396653416949",
    "SessionID": "gz1a74c25f17030396653416950",
    "DialogID": "",
    "Code": 0,
    "Message": "ok"
  },
  "Payload": {
    "StageInfo": "{\\"TrainResult\\":[\\"url1\\",\\"url2\\"],\\"RejectReason\\":\\"
    "Status": "MAKING",
    "StatusV2": "WAIT_PROCESS",
    "FailMessage": "",
    "StageV2": "CONFIRM"
  }
}
```

Result Code Description

Last updated：2024-07-18 18:08:10

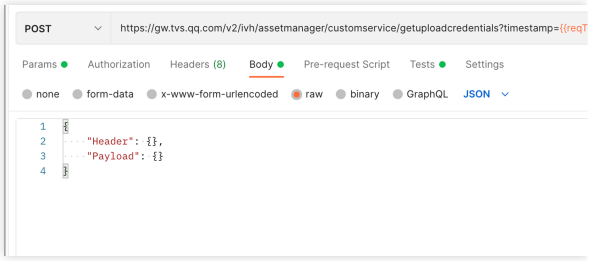
Code	Meaning	Measures
0	Success	No
100000	Internal system error	Retry or contact the administrator to locate the issue.
100001/100003	Request parameter errors (including incorrect parameter format and type, missing parameters, or lacking required parameters)	Check the parameter format and type according to the API protocol.
100002	Parameter value error	Check the parameter list according to the API protocol.
100005	Unauthorized operation	Check if the session has the necessary permissions for the request.
110006	The record does not exist.	Check if the taskid is correct.
100008	Exceed quota limit	The allocated quota has been used up.

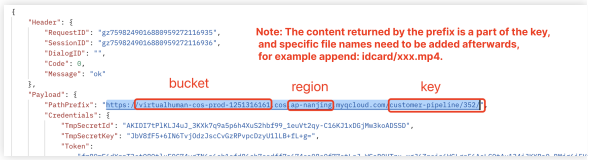
©2013-2025 Tencent Cloud International Pte. Ltd.

Interface Integration FAQs

Last updated：2024-07-18 18:12:11

Serial number	Problem description	Solutions	Remarks
1	Authentication failure occurs when the interface is called.	1. Ensure that the parameters are sorted in alphabetical order when generating the signature. 2. Ensure that the timestamp parameter is set to the current time. 3. See the example parameters on the right. Do the calculation according to your own signature steps, and when your result matches the example, it means that the signature calculation steps are correct.	The parameters are described as follows: appkey = e38267c0e86411ebb02aed82acb0ed99 accesstoken = f68f2d10ae9e4604b76fb05cf46bccec timestamp=1646636485 Signature result: fWuaC9kmaicCggXc693uK%2BsZQ8qe88O4HVQNT
2	A "miss Content Length" error occurs when the interface is called.	1. Check if the request body includes the header parameters and payload	

		parameters; there is an error without them, as shown in the example on the right. 2. The parameter types must be filled in strictly according to the interface documentation. 3. Check if the URL is spelled correctly.	
3	UnauthorizedOperation: unauthorized operation: No project was created for this appkey.	The customer has not been granted permission to use the image customized interface. Please contact Tencent personnel for permission.	
4	Do video materials need to be uploaded to the customer's Tencent Cloud COS?	Not required. Video materials will be sent to the Avatar's internal COS via a temporary token, so there is no need for the customer to activate Tencent Cloud COS.	
5	How to obtain the Bucket, Region, and	These can be parsed from the prefix, as	

	Key when the COS SDK is called?	shown in the example on the right.	

Video Generation Service API Documentation Overview

Last updated：2024-07-18 18:12:42

This documentation primarily describes the open API protocol for the **Tencent Cloud Broadcast Digital Human** aPaas platform.

API Calling Methods

Digital Human aPaas API Calling Method

Note:

After the appkey is purchased directly through Tencent Cloud billing, it can only be used to switch avatars within the selected avatar package and is not interchangeable between different avatar packages.

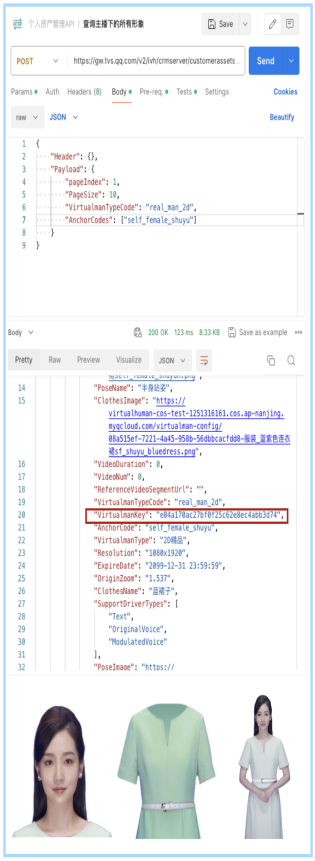
Access Process Diagram

Resource Query Logic Diagram

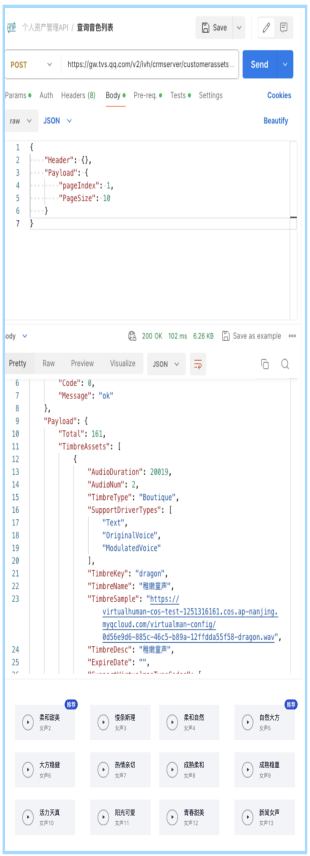
©2013-2025 Tencent Cloud International Pte. Ltd.

Personal asset management API - query the host

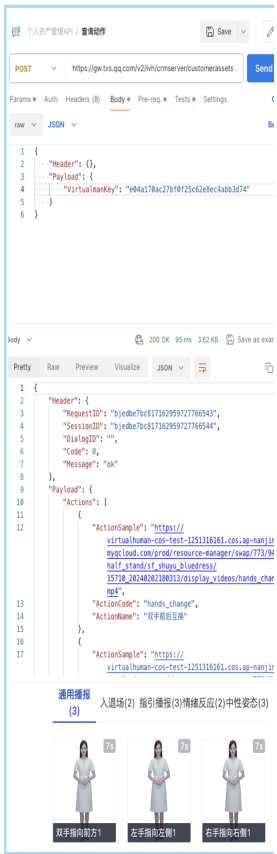
Personal asset management API - query all images under the host



Personal asset management API - query the timbre list



Personal asset management API - query the list of actions supported by the image



Digital Human aPaaS API Calling Methods

Last updated：2025-07-02 16:54:14

This documentation primarily describes the API call methods on the Tencent Cloud AI Digital Human aPaaS platform, including permissions, common parameters, and signature requirements.

I. API Calling Environment

Official environment: `https://gw-sg.tvs.qq.com`

II. Permission Application and Activation

Video generation aPaas API usage permissions require purchasing related services before usable. For initial stage testing, contact sales to apply for a free trial.

III. Methods for Requesting Common Parameters and Signature

Common Parameters

When calling any API on aPaaS, you need to include the following common parameters in the URL as QueryString:

Parameters	Type	Mandatory	Description
appkey	string	Yes	Fill in the appkey obtained after your permission application is approved.
timestamp	string	Yes	Request timestamp in seconds. The timestamp cannot differ from the current time by more than five minutes, otherwise, authentication will fail.
signature	string	Yes	Request signature (see the following Signature Method).

Signature Method

Request Parameter Signature Steps are as follows:

1. signature rules are as follows, example description below (for reference only).

```
appkey = e38267c0e86411ebb02aed82acb0ed99
accesstoken = f68f2d10ae9e4604b76fb05cf46bccec
```

```
Domain Name Routing = https://gw.tvs.qq.com/v2/ivh/videomaker/broadcastservice/vide
```

2. Sort all URL parameters except signature in alphabetical order, note that parameters in the body are not included, as the plaintext for generating the signature. Currently, only appkey and timestamp parameters, so the sorted and concatenated string example is:

```
appkey=e38267c0e86411ebb02aed82acb0ed99&timestamp=1646636485
```

3. Use the accesstoken to perform HmacSha256 encryption on the signature string, and then encode it with base64.

```
hashBytes = HmacSha256("appkey=e38267c0e86411ebb02aed82acb0ed99&timestamp=164663648
signature = Base64Encode(hashBytes)
```

4. The obtained signature value is:

```
BfWuaC9kmaicCggXc693uK+sZQ8qe88O4HVQNTdwZuo=
```

5. Urlencode the signature value (this is mandatory, otherwise it may cause intermittent authentication failures), and then concatenate it to get the final request URL as follows:

```
https://gw.tvs.qq.com/v2/ivh/videomaker/broadcastservice/videomake?appkey=e38267c0e
```

IV. Request and Response Structure

Request Body Structure

The request body is divided into Header and Payload. The Header can include RequestID to identify a single request for system issue tracking. Business parameters are placed in the Payload. Each API's input parameters in the product API documentation should be placed in the Payload for transmission.

```
{
    "Header": {
        "RequestID": "",
    },
    "Payload": {}
}
```

Response Body Structure

The response body is divided into two parameters: Header and Payload. The Header contains the result code, message, and the unique request ID which is the same as the value provided in the request body. The specific response parameters of the API will be placed in the **Payload**.

```
{
    "Header": {
```

```
        "Code": 0,
        "Message": "",
        "RequestID": "", // If not filled in during the request, it will be generat
    },
    "Payload": {}
}
```

Audio Production API

Last updated：2024-07-18 18:18:42

API Description

To preview the input text, you can query the timbre to be previewed through the Query Supported Timbres for VirtualmanKey API. Some avatars do not support changing the timbre.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/tts

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TimbreKey	string	No	Timbre key. When VirtualmanKey is empty, TimbreKey cannot be empty.
VirtualmanKey	string	No	Define information such as the role, clothing, pose, and resolution for the broadcast. The parameter is an enumerated value. When TimbreKey is empty, VirtualmanKey cannot be empty. By default, the first matching timbre for the avatar will be selected to produce the audio.
InputSsml	string	Yes	Text content to be broadcast which supports SSML tags with an upper limit of 20,000 characters (counted as Unicode characters)
Speed	float	Yes	Speech rate (1.0 is normal speed, with a range of [0.5 to 1.5]. A value of 0.5 represents the slowest speed, while 1.5 represents the fastest speed).
AudioStorageS3Url	string	No	A URL with an authenticated S3 protocol for storage can be provided, and the finished audio will be uploaded to that URL.

©2013-2025 Tencent Cloud International Pte. Ltd.

SampleRate	int	No	Sample rate which supports 24000 (24k) and 16000 (16k) with 24000 (24k) as the default
Codec	string	No	Audio format which supports mp3 and wav with mp3 as the default
SentenceMaxWords	int	No	The upper limit number of characters per sentence which ranges from 0 to 999. If 0 or nothing is provided, the default value is 30.
SentenceDisplayPunctuation	string	No	Punctuation marks to be displayed within sentences. Special character "0" means that no punctuation marks will be displayed, while "1" (default value) means that all punctuation marks will be displayed. You can also customize which punctuation marks to display.
SentenceSplitPunctuation	string	No	Punctuation marks for sentence segmentation with the default values being . ; ?! ... !?
Volume	int	No	Volume level, ranging from 0 to 10. The default is 0, which represents normal volume. The higher the values, the louder the volume.
EmotionCategory	string	No	Control the emotion of synthetic audio, and only multi-emotion timbres are supported for the use. See the Personal Asset Management API 4.5 Timbre List API for optional values.
EmotionIntensity	int	No	Controls the intensity of the synthesized audio emotion, with a range of [50,200]. This is only effective when EmotionCategory is not empty.

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The task ID for audio production. Use the taskId to access the <Audio and Video Production Progress Query API> to obtain the production progress and download link for the video.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "123",
        "InputSsml": "Hello, virtual anchor",
        "Speed": 1
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "TaskId": "123"
    }
}
```

Video Production API - Basic Edition

Last updated：2024-07-18 18:20:16

API Description

Use the SSML text and the digital human for video production. The final product video and subtitle file are returned through the Audio and Video Production Progress Query API.

Note:

Defining advanced parameters like anchor position is not supported. To use these features, switch to the Video Production API - Advanced Edition.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/videomake

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Define the broadcasting role, clothing, pose, and resolution. The parameter is an enumerated value. **Note:** You can query through the Query Image Asset Information - Query All Images under Anchors API.
InputSsml	string	Yes	The text content to be broadcast supports SSML tags. Refer to the Digital Human SSML Markup Language Specification for supported tag types and examples for tag usage. The content must not include line breaks, and symbols must be escaped. The upper limit is 20,000 characters (counted as Unicode characters). This field is required if DriverType is empty or set to Text.

SpeechParam	object	Yes	Define the detailed parameters of the audio.
SpeechParam.Speed	float	Yes	The speech rate (1.0 is normal speed, range [0.5-1.5]. A value of 0.5 indicates the slowest speed and a value of 1.5 indicates the fastest speed. Speech rate control is not effective when DriverType is set to audio-driven type).
SpeechParam.TimbreKey	string	No	Timbre key, and the avatar's own timbre is used by default.
SpeechParam.Volume	int	No	Volume level, ranging from 0 to 10. The default is 0, which represents normal volume. The higher the values, the louder the volume.
SpeechParam.EmotionCategory	string	No	Controls the emotion of the synthesized audio, supported only for multi-emotion timbres. See the Personal Asset Management API Paginated Query Timbre List for available values.
SpeechParam.EmotionIntensity	int	No	Controls the intensity of the synthesized audio emotion, with a range of [50,200]. This is only effective when EmotionCategory is not empty.
VideoParam	object	No	Define the detailed parameters for video synthesis.
VideoParam.Format	string	No	Video output format, and it is TransparentWebm by default. TransparentWebm: WebM format video with a transparent background GreenScreenMp4: MP4 format video with a green screen background
CallbackUrl	string	No	When the user adds a callback URL, the video production results will be sent as a POST request in a fixed format to that URL. For the fixed format, see Appendix II: Invocationback Request Body Format. Note: 1. The InvocationbackUrl length must be less than 1000 characters. 2. Only one request can be sent. Regardless of the issue causing the request to fail, it cannot be resent.
DriverType	string	No	Driver type, and it is Text by default.

			1. Text: It is text-driven and requires the InputSsml field to be filled. 2. OriginalVoice: It is original voice audio-driven and requires the InputAudioUrl field to be filled. 3. ModulatedVoice: It is modulated voice audio-driven and can specify timbre using Speech.TimbreKey. If not specified, the anchor's default timbre will be used.
InputAudioUrl	string	No	The audio URL to drive the digital human. This field is required when DriverType is OriginalVoice or ModulatedVoice. Audio format requirements: 1. For small sample avatars, the duration should not exceed 60 minutes and not be less than 0.5 seconds. For non-avatars, the duration should not exceed 10 minutes and not be less than 0.5 seconds. 2. Supported formats: WAV, MP3, WMA, M4A, and AAC.
VideoStorageS3Url	string	No	A URL with an authenticated S3 protocol for storage can be provided, and the finished video will be uploaded to that URL.
SubtitleStorageS3Url	string	No	A URL with an authenticated S3 protocol for storage can be provided, and the finished subtitles will be uploaded to that URL.
ConcurrencyType	string	No	The concurrency type used for video production tasks defaults to prioritizing dedicated concurrency, followed by shared policies. 1. Exclusive: Dedicated concurrency. If no dedicated concurrency is available, the task submission will fail. 2. Shared: Shared concurrency

Response Parameter

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The video production task ID. Use the TaskId to access the Audio and Video Production Progress Query API to obtain the production progress and results.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "123",
        "InputSsml": "Hello, I am the virtual <phoneme alphabet=\\"py\\" ph=\\"fu4\
        "SpeechParam": {
            "Speed": 1.0
        },
        "VideoParam": {
            "Format": "GreenScreenMp4"
        }
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "DialogID": "",
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "TaskId": "123"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Audio and Video Production Progress Query API

Last updated：2024-07-18 18:20:46

API Description

You can query the progress and results of a task using taskId. When the progress field in the return value reaches 100, you can use MediaUrl to obtain the download link for the final audio and video. For video production tasks, the SubtitlesUrl field will also be returned to retrieve the final SRT subtitle address. Audio and video resources without a custom storage URL are retained for only 7 days.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/getprogress

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
TaskId	string	Yes	The task ID is returned in the TTS Preview APIVideo Production API - Basic VersionVideo Production API - Advanced Version API.

Response Parameter

Parameters	Type	Mandatory	Description
Progress	int	Yes	Production progress ranges from -1 to 100: -1 indicates generation failure, and 100 indicates successful generation (reserved field, currently not for reference).
MediaUrl	string	Yes	Audio and video result address
SubtitlesUrl	string	No	The corresponding SRT subtitle address for the video is returned in video production.
Status	string	Yes	Production Status "COMMIT": Submitted and awaiting to queue "MAKING": In production "SUCCESS": Production succeeded

©2013-2025 Tencent Cloud International Pte. Ltd.

			"FAIL": Production failed
ArrayCount	int	Yes	The number of tasks queued before this task when the Status is "COMMIT"
FailCode	int	No	The error code returned when the production failed
FailMessage	string	No	Failure reasons returned when production fails which facilitate troubleshooting
TextTimestampResult	Array of [Sentence]	No	This field returns the text timestamp information for the TTS preview task.
Duration	int	Yes	Video duration, in milliseconds.

Sentence

Parameters	Type	Mandatory	Description
Sentence	string	Yes	Sentences derived from sentence splitting
Words	Array of [Word]	Yes	Information about each word in the sentence

Word

Parameters	Type	Mandatory	Description
Word	string	Yes	One character in the sentence
StartTimestamp	long	Yes	The starting time point of that character, and the value is divided by 10,000 represents milliseconds (equivalent to 0.1 microseconds).
EndTimestamp	long	Yes	The ending time point of that character, and the value is divided by 10,000 represents milliseconds (equivalent to 0.1 microseconds).

Request Sample

```
{
    "Header": {},
    "Payload": {
        "TaskId": 123
    }
}
```

Response Sample


```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Progress": 100,
        "MediaUrl": "url",
        "SubtitlesUrl": "",
        "ArrayCount": 0,
        "FailMessage": "",
        "Duration": 11810,
        "FailCode": 0,
        "TextTimestampResult": [
            {
                "Sentence": "How are you doing, virtual anchor?",
                "Words": [
                    {
                        "Word": "how",
                        "EndTimestamp": 6100000,
                        "StartTimestamp": 4500000
                    },
                    {
                        "Word": "are",
                        "EndTimestamp": 8200000,
                        "StartTimestamp": 6100000
                    },
                    {
                        "Word": "you",
                        "EndTimestamp": 9700000,
                        "StartTimestamp": 8200000
                    },
                    {
                        "Word": "doing",
                        "EndTimestamp": 11100000,
                        "StartTimestamp": 9700000
                    },
                    {
                        "Word": "virtual",
                        "EndTimestamp": 12900000,
                        "StartTimestamp": 11100000
                    },
                    {
```

```
                        "Word": "anchor",
                        "EndTimestamp": 16000000,
                        "StartTimestamp": 12900000
                    }
                ]
            }
        ]
    }
}
```

Video Production API - Advanced Version

Last updated：2025-03-28 16:42:54

API Description

Use ssml text and digital human for video production. The final video and subtitle file are returned through the Audio and video production progress query API. The advanced version of the API builds on the original by adding new resource parameters and expanding micro-editing capabilities. The supported features are detailed in the following table:

LOGO	Supporting adjusting the logo position (customizable X and Y axes)
	Supporting adjusting the logo size (scaling ratio)
Specifying a video background	In MP4 video files, a background image can be specified.
Anchors	Horizontal position adjustment
	Anchor size adjustment
	Anchor angle adjustment **Notes:** Only 3D avatars supported
Intro/outro video	A remote address to add video intro and outro can be specified.
Embedded subtitles	Turning embedded subtitles on or off

Calling Protocol

HTTPS + JSON
POST /v2/ivh/videomaker/broadcastservice/videomakeadvanced
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter	Type	Mandatory	Description

©2013-2025 Tencent Cloud International Pte. Ltd.

VirtualmanKey	string	Yes	Define the broadcasting role, clothing, pose, resolution, etc. The parameter is an enumerated value. **Notes:** You can query through the Query Image Asset Information - Query All Images under Anchors API.
InputSsml	string	No	The text content to be broadcast supports SSML tags. Refer to the Digital Human SSML Markup Language Specification for supported tag types and examples for tag usage. The content must not include line breaks, and symbols must be escaped. The upper limit is 20,000 characters (counted as Unicode characters). This field is required when DriverType is empty or set to Text.
SpeechParam	object	Yes	Define the detailed parameters of the audio.
SpeechParam.Speed	float	Yes	The speech rate (1.0 is normal speed, range [0.5-1.5]. A value of 0.5 indicates the slowest speed and a value of 1.5 indicates the fastest speed. Speech rate control is not effective when DriverType is set to audio-driven type).
SpeechParam.TimbreKey	string	No	Timbre key, and the avatar's own timbre is used by default.

SpeechParam.Volume	int	No	Volume level, ranging from 0 to 10. The default is 0, which represents normal volume. The higher the values, the louder the volume.
SpeechParam.EmotionCategory	string	No	Control the emotion of synthetic audio, and only multi-emotion timbres are supported for the use. See the Personal Asset Management API Paginated Query Timbre List API for optional values.
SpeechParam.EmotionIntensity	int	No	Controls the intensity of the synthesized audio emotion, with a range of [50,200]. This is only effective when EmotionCategory is not empty.
VideoParam	object	No	Define the detailed parameters for video synthesis.
VideoParam.Format	string	No	Video output format; default value: MP4 TransparentWebm: Transparent background WebM format video, supporting some micro-editing capabilities (anchor parameters supported) GreenScreenMP4: Green screen MP4 format video, not supporting micro-editing capabilities MP4: MP4 format video supporting micro-editing capabilities
VideoParam.BackgroundFileUrl	string	No	Video background image/video download

			path, supporting jpg, png, and MP4 formats. The image/video resolution must match the video resolution. If not provided, the default is a green-screen video. The file size limit is 500 MB.
VideoParam.VideoHeadFileUrl	string	No	Intro video which supports MP4 format. The resolution must match the video resolution, with a file size limit of 500 MB.
VideoParam.VideoTailFileUrl	string	No	Outro video which supports MP4 format. The resolution must match the video resolution, with a file size limit of 500 MB.
VideoParam.ShowSubtitles	boolean	No	Whether to display subtitles in the video. By default, subtitles are not displayed. Enabling subtitles will significantly increase the video production time.
VideoParam.SubtitlesParam	object	No	Define parameters for how subtitles are displayed in the video.
VideoParam.SubtitlesParam.MaxWords	int	No	The upper limit of characters displayed per page of subtitles, with a range from 0 to 999. The default value is 0. Default display rule: subtitles are shown within 80% of the video width; if exceeded, the text is paginated.
VideoParam.SubtitlesParam.DisplayPunctuation	string	No	Punctuation marks to be displayed in the subtitles. The special character "0"

			indicates no punctuation will be displayed, while "1" (the default value) indicates all punctuation will be displayed. You can also customize which punctuation marks to display by specifying them.
VideoParam.SubtitlesParam.SplitPunctuation	string	No	Punctuation marks that require subtitles to paginate, with the default values being: . ; ? ! ... !?
VideoParam.LogoParams	Array of [LogoParam]	No	Define parameters related to the logo in the video.
VideoParam.SmartActionEnabled	bool	No	Whether to enable intelligent actions. The default is disabled. Effective conditions: DriverType=Text and InputSsml does not contain action tags.
VideoParam.AnchorParam	object	No	Define parameters related to the anchor in the video.
VideoParam.AnchorParam.HorizontalPosition	float	No	Define the anchor's horizontal position (0 is the center). The Tag effect varies for different anchors: Basic: Supporting left and right movements Standard: Supporting left and right movements Advanced: Not supporting left and right movement; any value will be treated as 0. **Notes:** You can query the tag value through the Query Image Asset Information - Query Anchor API.

VideoParam.AnchorParam.VerticalPosition	float	No	Define the anchor's vertical position (0 is the center). The Tag effect varies for different anchors. Basic: Supports upward and downward movements Standard: Only supports downward movement (>=0). If a value less than 0 is provided, it defaults to 0. Advanced: Only supports downward movement (>=0). If a value less than 0 is provided, it defaults to 0. **Notes:** You can query the tag value through the Query Image Asset Information - Query Anchor API.
VideoParam.AnchorParam.Scale	float	No	Anchor size (1 is the default size, range (0,1]). Anchors with the Basic tag can have a size greater than 1. Avatar fall under Basic.
VideoParam.AnchorParam.Angle	int	No	Anchor angle (default is 0 degrees, and the range is [0,360]). The effect varies based on the anchor's tag. Basic: Supported only for 3D anchors Standard: Not supported Advanced: Not supported
VideoParam.AnchorParam.AnchorExtraParam	string	No	Additional configurable parameters for anchors currently only include the 3D clothing color change parameters. The parameters that can be configured vary by anchor. For details, see the SupportAnchorExtraParam parameter in the Query

			Image Asset Information - Query all images of the Anchor API; This parameter should be organized in JSON string format. See the request sample to organize the JSON.
VideoParam.SmallSampleParam	object	No	Define special parameters related to avatars. This parameter is not effective for non-avatars.
VideoParam.SmallSampleParam.MakeType	string	No	Define Avatar production type: Default: The default configuration. Production will starts with a random starting frame. StartIdx and EndIdx do not take effect. Custom: Specify a video segment by filling in StartIdx and EndIdx to select the segment. The default generated video will loop back and forth using this "specified video segment". Circle: The starting and ending frames align. You can fill in StartIdx to specify the starting frame of the video (EndIdx is not effective). Note: This type is only used for audio-driven videos.
VideoParam.SmallSampleParam.StartIdx	int	No	Starting frame number, effective when MakeType is set to Custom or Circle. If it is filled in, the generated video will start from this frame; if it is not filled in, the video will start

			from the default frame number, which is frame 0.
VideoParam.SmallSampleParam.EndIdx	int	No	Ending frame number, effective when MakeType is set to Custom. If it is specified, the generated video will end at this frame; if it is not specified, the video will end at the default frame number, which is the end of the selected video segment.
CallbackUrl	string	No	When the user adds a callback URL, the video production results will be sent in a fixed format as a POST request to that URL. For the fixed format, see Appendix II: Callingback Request Format. Note: 1. The InvocationbackUrl length must be less than 1000 characters. 2. Only one request will be sent. Regardless of the issue causing the request to fail, it cannot be resent.
DriverType	string	No	Driver type, and it is Text by default. 1. Text: It is text-driven and requires the InputSsml field to be filled. 2. OriginalVoice: It is original voice audio-driven and requires the InputAudioUrl field to be filled. 3. ModulatedVoice: It is modulated voice audio-driven and can specify timbre using Speech.TimbreKey. If not

			specified, the anchor's default timbre will be used.
InputAudioUrl	string	No	The audio URL to drive the digital human. This field is required when DriverType is OriginalVoice or ModulatedVoice. Audio format requirements: 1. For avatars, the duration should not exceed 60 minutes and not be less than 0.5 seconds. For non-avatars, the duration should not exceed 10 minutes and not be less than 0.5 seconds. 2. Supported formats: WAV, MP3, WMA, M4A, and AAC.
VideoStorageS3Url	string	No	A URL with an authenticated S3 protocol for storage can be provided, and the finished video will be uploaded to that URL.
SubtitleStorageS3Url	string	No	A URL with an authenticated S3 protocol for storage can be provided, and the finished subtitles will be uploaded to that URL.
ConcurrencyType	string	No	The concurrency type used for video production tasks defaults to prioritizing dedicated concurrency, followed by shared policies. 1. Exclusive: Dedicated concurrency. If no dedicated concurrency is available, the task submission will fail.

©2013-2025 Tencent Cloud International Pte. Ltd.

			2. Shared: Shared concurrency

LogoParam

Parameter	Type	Mandatory	Description
LogoFileUrl	string	No	Logo image file download path, supporting jpg and png formats.
PositionX	int	No	X-coordinate of the logo image's top-left corner (coordinate range depends on the video resolution)
PositionY	int	No	Y-coordinate of the logo image's top-left corner (coordinate range depends on the video resolution)
Scale	float	No	Logo image scaling ratio (1.0 represents the original image size.)

VideoParam.Anchor.AnchorExtraParam example

Notes:

ColorValue conversion rule: Convert the hexadecimal color value to decimal after removing the # symbol, e.g. #FAFAFA => 16448250.

```
{
    "shirtColor": {
        "colorValue": 10491928
    },
    "clotheColor": {
        "colorValue": 10491928
    },
    "shoeColor": {
        "colorValue": 10491928
    }
}
```

Response Parameter

Parameter	Type	Mandatory	Description

TaskId	string	Yes	The video production task ID. Use the taskId to access the Audio and Video Production Progress Query API to obtain the production progress and results.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "123",
        "InputSsml": "Hello, I am the virtual <phoneme alphabet=\\"py\\" ph=\\"fu4\
        "SpeechParam": {
            "Speed": 1
        },
        "VideoParam": {
            "Format": "Mp4",
            "BackgroundFileUrl": "url1",
            "VideoHeadFileUrl": "url2",
            "VideoTailFileUrl": "url3",
            "ShowSubtitles": true,
            "LogoParams": [
                {
                    "LogoFileUrl": "http://virtualhuman-cos-test-1251316161.cos.ap-
                    "PositionX": 1561,
                    "PositionY": 751,
                    "Scale": 1.0
                }
            ],
            "AnchorParam": {
                "HorizontalPosition": 0.5,
                "Scale": 1.0,
                "AnchorExtraParam": "\\"shirtColor\\": {\\"colorValue\\": 10491928}
            }
        }
    }
}
```

Response Sample

```
{
   "Header": {
```

```
        "Code":0,
        "DialogID":"",
        "Message":"",
        "RequestID":"123",
    },
    "Payload":{
        "TaskID":"123"
    }
}
```

Customer Resource Query Anchor API

Last updated：2024-07-18 18:21:31

API Description

Query all anchors owned by customers within their permissions and anchor avatars through AppKey and AccessToken.

Note:

This API is gradually being phased out. Use the Query Image Asset Information - Query Anchor API for queries.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/getanchor

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
GetAllResource	bool	Yes	When set to false, it retrieves resources specific to the current AppKey; when set to true, it retrieves all resources associated with the current user.
VirtualmanTypeCode	string	No	Digital human type code: real_man_2d: 2D Premium Anchor real_man_3d: 3D Anchor

Response Parameter

Parameters	Type	Mandatory	Description
VirtualmanResources	Array of [VirtualmanResource]	Yes	Resources of the digital human within customers' permissions

©2013-2025 Tencent Cloud International Pte. Ltd.

VirtualmanResource

Parameters	Type	Mandatory	Description
AppKey	string	Yes	Requested AppKey
Virtualmans	Array of [Virtualman]		Resources of the digital human corresponding to this AppKey

Virtualman

Parameters	Type	Mandatory	Description
AnchorName	string	Yes	Anchor Name
AnchorCode	string	Yes	Anchor Code
HeaderImage	string	Yes	Digital human avatar image URL
VirtualmanType	string	Yes	Digital human type
VirtualmanTypeCode	string	Yes	Digital human type code
Tag	string	Yes	Digital human model tags, categorized into three types: 1. Basic 2. Standard 3. Advanced Currently, the tag only affects the horizontal and vertical positioning of anchors. See the API Video Production API - Advanced Version.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "GetAllResource": true,"VirtualmanTypeCode": "real_man_2d"
    }
}
```

Response Sample


```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "VirtualmanResources": [
            {
                "AppKey": "22222",
                "Virtualmans": [
                    {
                        "AnchorName": "Yani",
                        "HeaderImage": "url",
                        "AnchorCode": "yani","VirtualmanTypeCode": "real_man_2d",
                        "VirtualmanType": "2D Real Person"
                    }
                ]
            }
        ]
    }
}
```

Querying All Images of a Specific Anchor

Last updated：2024-07-18 18:21:45

API Description

Query all images under a specific anchor code, including all the anchor's clothing, poses, resolutions, and VirtualmanKeys.

Note:

This API is gradually being phased out. Use the Query Image Asset Information - Query all images under the anchor API for queries.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/getresourcebyanchor

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
GetAllResource	bool	Yes	When set to false, it retrieves resources specific to the current AppKey; when set to true, it retrieves all resources associated with the current user.
AnchorCode	string	Yes	Anchor code, which can be obtained through the Customer Resource Query Anchor API

Response Parameter

Parameters	Type	Mandatory	Description
VirtualmanResources	Array of [VirtualmanResource]	Yes	Resources of the digital human within customers' permissions

©2013-2025 Tencent Cloud International Pte. Ltd.

VirtualmanResource

Parameters	Type	Mandatory	Description
AppKey	string	Yes	Requested AppKey
Virtualmans	Array of [Virtualman]	Yes	Digital human under this role code

Virtualman

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Define the broadcasting role, clothing, pose, and resolution.
Name	string	Yes	Role name
Code	string	Yes	Role code
ClothesName	string	Yes	Digital human clothing
PoseName	string	Yes	Digital human pose
Resolution	string	Yes	Digital human resolution
HeaderImage	string	Yes	Digital human avatar image URL
PoseImage	string	Yes	Digital human pose image URL
ClothesImage	string	Yes	Digital human clothing image URL
SupportDriverTypes	Array of [string]	Yes	Supported drive types of the digital human 1. Text: Text-driven 2. OriginalVoice: Original voice audio-driven 3. ModulatedVoice: Modulated voice audio-driven
SupportAnchorExtraParams	Array if [SupportAnchorExtraParam]	Yes	Description of configurable parameters for the digital human avatars: Currently only 3D clothing color change parameters are included.
OriginZoom	String type floating point	Yes	Initial scaling factor of the

	number		digital human, accurate to three decimal places

Request Sample

```
{
    "Header": {},
    "Payload": {
        "GetAllResource": true,
        "AnchorCode": "yani"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "VirtualmanResources": [
            {
                "AppKey": "22222",
                "Virtualmans": [
                    {
                        "VirtualmanKey": "abddd",
                        "AnchorName": "Yani",
                        "AnchorCode": "yani",
                        "ClothesName": "Yellow Suit",
                        "PoseName": "Standing",
                        "Resolution": "1920x1080",
                        "HeaderImage": "url",
                        "PoseImage": "url",
                        "ClothesImage": "url",
                        "SupportDriverTypes": [
                            "Text",
                            "OriginalVoice"
                        ],
                        "SupportAnchorExtraParams": [
```

```
                                {
                                    "Key": "shirtColor",
                                    "Description": "Suit Color",
                                    "Type": 1,
                                    "Value": [
                                        "colorValue"
                                    ]
                                },
                                {
                                    "Key": "clotheColor",
                                    "Description": "Shirt Color",
                                    "Type": 1,
                                    "Value": [
                                        "colorValue"
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
}
```

Querying the Supported Timbres for VirtualmanKey (to Be Deprecated)

Last updated：2024-07-18 18:22:00

API Description

Query the supported timbres for a given VirtualmanKey based on the VirtualmanKey.

Note:

The current avatar and timbre have been unbound. The timbres retrieved using the Paginated Query Timbre List can now be used with any avatar.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/gettimbre

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Unique identifier for the avatar, which can be obtained through the Query All Avatars under an Anchor API.

Response Parameter

Parameters	Type	Mandatory	Description
Timbres	Array of [Timbre]	Yes	Supported Timbre List

Timbre

Parameters	Type	Mandatory	Description
TimbreKey	string	Yes	Digital human timbre code
TimbreName	string	Yes	Digital human timbre name
TimbreSample	string	Yes	Digital human timbre demo sample URL
TimbreDesc	string	Yes	Digital human timbre description

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "2803ca3b5ec64260a2c11e396eaac519"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Timbres": [
            {
                "TimbreKey": "female_1",
                "TimbreName": "female voice 1",
                "TimbreSample": "url"
            }
        ]
    }
}
```

Querying the Supported Actions for VirtualmanKey

Last updated：2024-07-18 18:22:11

API Description

Query the supported actions for a given VirtualmanKey based on the VirtualmanKey.

Note:

This API is gradually being phased out. Use the Query Image Supported Actions List API to retrieve the supported actions.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/videomaker/broadcastservice/getaction

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Unique identifier for the avatar, which can be obtained through the Query All Avatars under an Anchor API.

Response Parameter

Parameters	Type	Mandatory	Description
Actions	Array of [Action]	No	List of supported actions

Action

Parameters	Type	Mandatory	Description
ActionName	string	Yes	Action name of the digital human
ActionCode	string	Yes	Action code of the digital human, corresponding to SSML action tags
ActionSample	string	Yes	Example URL for the digital human actions

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "2803ca3b5ec64260a2c11e396eaac519"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Actions": [
            {
                "ActionName": "Yunxuan",
                "ActionCode": "code1",
                "ActionSample": "url"
            }
        ]
    }
}
```

Appendix

Appendix I: Result Code Dictionary

Last updated：2024-07-18 18:22:24

I. API Error Codes

Code	Meaning	Measures
0	Success	No
-101000	Internal system error	Retry or contact the administrator to locate the issue.
-101001	Computing power is insufficient.	Retry or contact the administrator to locate the issue.
-101002	Image downloading failed.	Check whether the provided image URL can be accessed and downloaded.
-101003	The request for S3 failed.	Retry or contact the administrator to locate the issue.
100001/100003	Request parameter errors (including incorrect parameter format and type, missing parameters, or lacking required parameters)	Check the parameter format and type according to the API protocol.
100002	Parameter value error	Check the parameter list according to the API protocol.
100005	Unauthorized operation	Check if the session has the necessary permissions for the request.
100008	Exceeded quota limit	The session exceeded the concurrency limit.
100009	The resource does not exist.	Check whether the current permission scope is exceeded.
110005	The record already exists.	Repeated requests are excessively rapid.
110006	The record does not exist.	Check whether the requested session exists.
110007	The virtualmankey has expired.	Check the validity period of the key

©2013-2025 Tencent Cloud International Pte. Ltd.

		associated with the root account.
100013	No valid hourly package was found or insufficient hourly package balance.	Request or purchase the appropriate hourly package.
100015	The digital human avatar has expired.	Apply for an extension of the avatar.
100016	No avatar assets were found.	Check whether the VirtualmanKey being used is valid.
801000	The calling remote service has an exception.	Retry or contact the administrator to locate the issue.
801002	The remote service call returned an abnormal status code.	Retry or contact the administrator to locate the issue.
801004	Operations are too frequent. Please try again later.	Avoid submitting multiple tasks at the same time.
801005	The text failed to pass the security verification.	Check whether the SSML contains sensitive information.

II. Audio and Video Production Result Error Codes

Code	Meaning	Measures
0	Success	No
801000	Internal system error	Retry or contact the administrator to locate the issue.
801001	Database operation error	Retry or contact the administrator to locate the issue.
801002	Error in calling remote service	Retry or contact the administrator to locate the issue.
801003	Task production exceeded the retry limit.	Retry or contact the administrator to locate the issue.
801004	Task production encountered a crash.	Retry or contact the administrator to locate the issue.
801006	Operation on S3 failed.	Retry or contact the administrator to locate the issue.

801007	Task execution timeout.	Retry or contact the administrator to locate the issue.
801009	Parameter error	Check whether the parameters in FailMessage meet the API requirements.
801010	Retrieving the URL resource failed.	Check whether the provided URL resource can be accessed and downloaded.
801012	Task parameter parsing failed.	Retry or contact the administrator to locate the issue.
801014	Downloading subtitle files/downloading generated video failed.	Retry or contact the administrator to locate the issue.
801015	Uploading files to the specified S3 failed.	Check whether the provided VideoStorageS3Url and SubtitleStorageS3Url are available.
801401	Audio production. Parsing task parameters in audio production failed.	Retry or contact the administrator to locate the issue.
801402	Audio production. Creating local files in audio production failed.	Retry or contact the administrator to locate the issue.
801403	Audio production. Receiving audio data in audio production failed.	Retry or contact the administrator to locate the issue.
801404	Audio production. A business exception occurred while receiving audio data in audio production.	Retry or contact the administrator to locate the issue.
801405	Video production. Parsing task parameters in video production failed.	Retry or contact the administrator to locate the issue.
801503	Video production. Invocationing remote services in video production failed.	Retry or contact the administrator to locate the issue.
801507	Video production. Converting the resolution of input images or other resources in video production failed.	Retry or contact the administrator to locate the issue.
801508	Video production. Obtaining the number of frames for a specified Avatar video clip in video production failed.	Retry or contact the administrator to locate the issue.
801509	Video production. Check whether the number of input frames in a Avatar	Check whether StartIdx and EndIdx conform to the API definition.

	conforms to requirements in video production.	
801510	Video production. The audio driver file duration verification failed in video production.	Check the audio drive file, retry, or contact the administrator to locate the error.

Appendix II: Callback Request Body Format

Last updated：2024-07-18 18:22:38

Parameters	Type	Mandatory	Description
Progress	int	Yes	Production progress: -1 indicates generation failure, and 100 indicates successful generation (reserved field, currently not for reference).
MediaUrl	string	Yes	Audio and video result address
SubtitlesUrl	string	No	The corresponding SRT subtitle address for the video is returned in video production.
Status	string	Yes	Production Status "SUCCESS": Production succeeded "FAIL": Production failed
FailMessage	string	No	Failure reasons returned when production fails which facilitate troubleshooting
TaskId	string	Yes	The taskId returned from the video production request

Request Sample

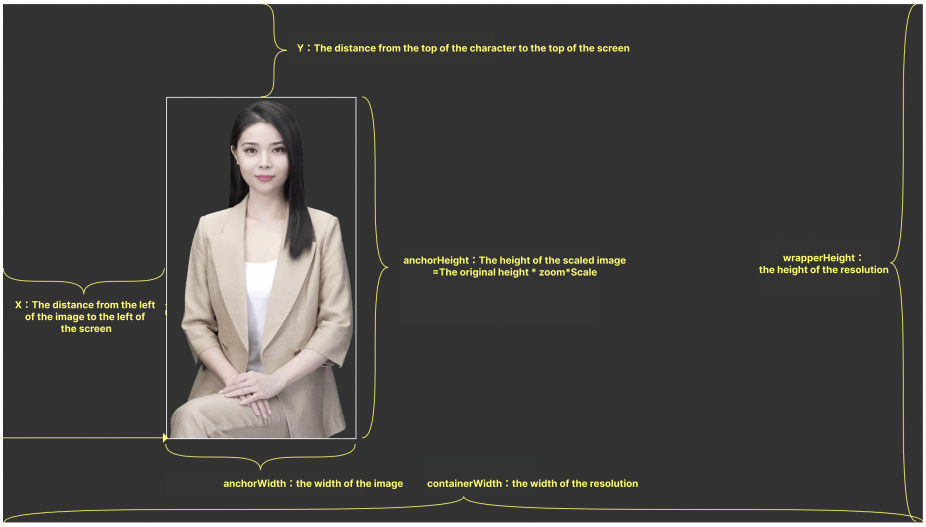
```
{
   "Payload": {
       "Progress": 100,
       "MediaUrl": "XXX",
       "SubtitlesUrl":"XXX",
       "Status": "SUCCESS",
       "FailMessage": "",
       "TaskId":"XXX"
    }
}
```

API Integration FAQs

Last updated：2024-07-18 18:22:51

Problem Description	Solutions
Authentication failure occurs when the API is called.	1. Ensure that the parameters are sorted in alphabetical order when generating the signature. 2. Ensure that the timestamp parameter is set to the current time. 3. See the example parameters on the right. Do the calculation according to your own signature steps, and when your result matches the example, it means that the signature calculation steps are correct.
Parameter errors occur when the API is called.	1. Check whether the request body includes the header parameters without which an error occurs, as shown in the example on the right. 2. The parameter types must be filled in strictly according to the API documentation.
The quantity of avatars found is incorrect, or no avatars are found.	If you have subscribed to Avatar, see the Paginated Query Avatar List API to query. If you have subscribed to 2D premium avatars, see the Customer Resource Query Anchor API for the query. **Note:** After ordering the hourly broadcast package on the official website, if both Avatar and 2D premium avatars are enabled, you need to see both sets of APIs to query the relevant avatars separately.
Does it support generating videos with a	Yes, it supports generating videos with a transparent background in WebM format.

©2013-2025 Tencent Cloud International Pte. Ltd.

transparent background?	
Why does the WebM video with a transparent background play have a green screen when it is played?	It indicates that the video player used by the customer does not support the WebM format. Try to play the video in Chrome; if it appears with a black background, it means the video itself is fine.
Avatar position parameters are not effective after adjustment.	1. Check whether the API being called is the Basic version or the Advanced version, as only the Advanced version supports this feature. 2. Check whether the video output format is set to GreenScreenMP4. Position adjustments are not supported for this format.
Anchor Position Calculation Method	 **Parameter 1: HorizontalPositon: The horizontal position of the host (the default value is 0, a negative number means to adjust to the left, and a positive number means to adjust to the right).** Definition: The distance between the central axis of the screen and the central axis of the image / the width of the entire screen. Formula: HorizontalPosition = (x + anchorWidth / 2 - containerWidth / 2) / containerWidth **Parameter 2: verticalposition: The vertical position of the host (the default value is 0, a negative number means to adjust upwards, and a positive number means to adjust downwards).** Definition: The distance between the bottom of the image and the bottom of the screen / twice the height of the image. Formula: VerticalPosition = (anchorHeight + y - wrapperHeight) / 2 / anchorHeight **Parameter 3: Scale: The size of the host (the default value is: 1; the smaller the value, the smaller the host).** Scaling rule: The lower left vertex remains motionless, and scales proportionally according to the width and height of the image. The definition of the related parameters: 1.AnchorHeight: That is the image height with the superimposed scaling coefficient, which is "the original picture height" * originzoom * scale 2.Anchorwidth: That is the image width with the superimposed scaling coefficient, which is "the original picture width" * originzoom * scale 3.Original picture height and width: That is, in the "Personal Asset Management APAaaS API Agreement" → 4.1 Pagination Query Avatar Image List API, the height and width of the PoseImage 4.Originzoom: That is the default scaling coefficient. In order to ensure that the characters have the best display effect by default under different resolutions, a default scaling coefficient is given. It can be queried in the "Personal Asset Management APAaaS API Agreement" → 4.1 Pagination Query Avatar Image List API.
How to calculate the initial size of	The digital human query API returns OriginZoom (original zoom factor) and PoseImage (digital human pose image). The initial size of the digital human is calculated by multiplying OriginZoom by the width and height of PoseImage.

the digital human?	
Action insertion in the text does not work.	You need to reserve a certain amount of text before and after each action. Since each action requires a different amount of reserved text, it is recommended to reserve at least 10 characters before and after each action.
Does the digital human support adjustments for the speech rate, intonation, and polyphonic characters?	Sure, see the SSML tags section in the API documentation, i.e. Appendix II: SSML Specification.
The specified external S3 storage did not receive the digital human video upload.	Please check if the S3 permission settings allow external writing, or if the S3 link in the parameters includes authentication information.
There are no subtitle files.	Subtitle files are only available for text-driven videos; they are not available for audio-driven videos.
Quota exceeded: The number of audio production submissions exceeds the limit: 1.	By default, only one audio production concurrency is enabled when the permission is granted, meaning only one audio task can be in production at a time. You must wait for the current task to be completed before submitting another. This can be resolved by purchasing additional audio concurrency.
Quota exceeded: The quantity of video production submissions	1. For each avatar type, each user is allowed to process up to 5 video production tasks concurrently by default, shared between the main and sub-accounts. These 5 tasks share concurrency resources with all customers, which may result in task queuing. 2. You can purchase additional dedicated concurrency for broadcasts. For example, if you buy 1 additional dedicated concurrency, you can process up to 6 video production tasks simultaneously. This additional concurrency is exclusive to the user who purchased it.

exceeds the limit: n.	Solution to the issue: 1. Users can implement a task queuing policy. 2. Purchase dedicated concurrency.

Interactive Digital Human Service API Documentation
Overview

Last updated：2025-04-09 10:24:55

This documentation primarily describes the open API protocol for the **TCADH Interaction** aPaas platform.

Must-Know

Before calling this API, please make sure you have purchased products related to the avatar.

API Calling Methods

When calling each API, you need to pass in common parameters and a signature in the query of the API. For an explanation of the parameters and signature methods, see the following link:
Avatar aPaas Interface Calling Method

Interactive Digital Human API Overview

Function Category	API Title		No.	API
Session Management	Create a session	Create a session using the Digital Human Platform project	1	POST /v2/ivh/sessionmanager/sessionmanagerservic
		Create a session	2	POST /v2/ivh/sessionmanager/sessionmanagerservice/creat

		using the image asset		
	Query Session Status		3	POST /v2/ivh/sessionmanager/sessionmanagerservic
	Starting Session		4	POST /v2/ivh/sessionmanager/sessionmanagerservic
	Close session		9	POST /v2/ivh/sessionmanager/sessionmanagerservic
	Query session list	Query session list under uin	-	POST /v2/ivh/sessionmanager/sessionmanagerservice/listse

		Query the conversation list under the Digital Human Project	-	POST /v2/ivh/sessionmanager/sessionmanagerservice/listse
		Query the conversation list under the personal asset image	-	POST /v2/ivh/sessionmanager/sessionmanagerservice/listse
Instruction-driven digital human Note: Instruction-driven application scenarios. See instruction-driven.	Persistent connection drive (WebSocket)	Create a long connection channel	5	WSS /v2/ws/ivh/interactdriver/interactdriverservice/co
		Text-driven instructions (WebSocket)	6	-
		Streaming Text-driven Instructions (WebSocket)	6	-

		Audio-driven Instructions (WebSocket)	6	-
		Heartbeat Instruction (WebSocket)	7	-

	Direct drive (HTTPS)	Text-driven Instructions (HTTP non-persistent connection)	8	POST /v2/ivh/interactdriver/interactdriverservice/comm
Other APIs	Real-time update of digital human visual parameters		-	POST /v2/ivh/sessionmanager/sessionmanagerservice/upda

	Obtain the key to enter the TRTC room	-	POST /v2/ivh/sessionmanager/sessionmanagerservic

Access Process

Notes:

The serial number corresponding to that in the flowchart corresponds to the serial number in the API table above.

Process Description

Get the interactive digital human service by following the steps below:

1. Purchase related products and services: purchase link;

2. Create a session: Use one of the following two methods to create a session with the cloud, and obtain the session status and playback URL of the digital human video.

Create a session using the Digital Human Platform project

Create a session using the image asset

3. Wait for the stream to be ready: When the session status is preparing, it means the digital human model is loading. The duration generally does not exceed two minutes. By calling the Query Session Status API in a loop, wait until the session status is in progress. For the status code explanation of the session status, see Query Session Status.

4. Play video stream;

5. Start a session: After the video stream is ready, you need to call the start session API to enable the digital human to be driven by instructions.

6. Send drive instructions: For descriptions and requirements of various drive instructions, see instruction sending requirements.

7. Close a session: Please close the session after the interaction is completed.

Digital Human aPaaS API Calling Methods

Last updated：2025-04-09 10:04:01

This documentation primarily describes the API call methods on the Tencent Cloud AI Digital Human aPaaS platform, including permissions, common parameters, and signature requirements.

API Invocation Environment

Official Environment: `https://gw-sg.tvs.qq.com`
Official Environment: `wss://gw-sg.tvs.qq.com`

Permission Application and Activation

You can activate the interactive digital human apass API usage permissions through Place Orders on the Official Website.

2.1. Obtaining Necessary API Parameters

To call the interactive digital human aPass API, you must carry the appkey, accesstoken, and virtualmankey corresponding to the digital human project. The specific acquisition method can be found in the following interface of the configuration platform:
The log-in address for the following interface is `https://xiaowei.cloud.tencent.com/ivh#/application` . Please log in using your Tencent Cloud account and create a project. Follow the instructions in the screenshot to obtain the appkey, accesstoken, and virtualmankey.

Requesting Common Parameters and Signature Methods

3.1. Common Parameters

When calling any API on aPaaS, you need to include the following common parameters in the URL as QueryString:

Parameters	Type	Mandatory	Description
appkey	string	Yes	Use the appkey obtained from "2.1. Obtaining Necessary API

			Parameters".
timestamp	string	Yes	Request timestamp in seconds. The timestamp cannot differ from the current time by more than five minutes, otherwise, authentication will fail.
signature	string	Yes	Request signature (see section 3.2 for signature calculation)

3.2. Signature Method

Request Parameter Signature Steps are as follows:

1. The signature rules are as follows. This is an example for reference only.

```
appkey = e38267c0e86411ebb02aed82acb0ed99
accesstoken = f68f2d10ae9e4604b76fb05cf46bccec
domain name routing = https://gw.tvs.qq.com/v2/ivh/sessionmanager/sessionmanagerser
```

2. Sort all parameters except signature in **lexicographical order** as the plaintext for generating the signature. Currently, the HTTPS API only has two parameters: appkey and timestamp. Therefore, the sorted and concatenated string example is:

```
appkey=e38267c0e86411ebb02aed82acb0ed99&timestamp=1646636485
```

Notes:

The url for the API of create long connection channel will have three parameters: appkey, requestid, and timestamp. Note that all of them need to be concatenated.

Concatenation example: appkey=xxx&requestid=xxxx&timestamp=xxxxx

3. Encrypt the plaintext for generating the signature using accesstoken with HmacSha256, and then perform base64 encoding.

```
hashBytes = HmacSha256("appkey=e38267c0e86411ebb02aed82acb0ed99&timestamp=164663648
signature = Base64Encode(hashBytes)
```

4. Get the signature value:

```
BfWuaC9kmaicCggXc693uK+sZQ8qe88O4HVQNTdwZuo=
```

5. Perform urlencode on the signature value (must perform URL encoding, otherwise intermittent authentication failure will occur), and then concatenate to get the final request URL as follows:

```
https://gw.tvs.qq.com/v2/ivh/sessionmanager/sessionmanagerservice/createsession?app
```

IV. Request and Response Structure

Request Body Structure

The request body is divided into two parameters: Header and Payload. There is no need to transmit parameters in the Header, and the business parameters for each API are transmitted in the Payload.

Notes:

Header and Payload must be included in the request body. They are essential.

```
{
    "Header": {
    },
    "Payload": {
      #Business parameter
    }
}
```

Response Body Structure

The response body is divided into two parameters: Header and Payload. The Header contains the result code, message, and the unique request ID. The specific response parameters of the API will be placed in the **Payload**.

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "", // generated and returned by the cloud
    },
    "Payload": {}
}
```

Live Session Management
Creating a New Live Stream Session

Last updated：2025-01-17 16:23:25

It is used to create a new video stream and obtain the playback URL for the digital human video.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/createsession

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
SessionId	String	No	A unique identifier for the session. The SaaS environment does not allow input from the cloud; the Private Environment allows input.
VirtualmanProjectId	String	Yes	The ID for the digital human project. (It is the same as the previous VirtualmanKey.)
UserId	String	Yes	The unique identifier for the user, maintained by the caller. Creating a new stream with the same UserIdentifier will cause the previous stream with that UserIdentifier to close.
Protocol	String	Yes	Currently supported parameters: rtmp, trtc and webrtc
DriverType	int	Yes	The driving method for the digital human. 1: text-only driving; 3: audio driving (original voice), which supports both audio and text driving modes in this setting;
ProtocolOption	ProtocolOption	No	Custom parameters for the protocol; ignore trtc if it

			is not used.

ProtocolOption

Name	Type	Required	Description
TrtcUseExternalApp	boolean	No	It indicates whether to use an external trtc. If you do not use it, th trtc appid of the digital human platform will be used. **Notes:** When the unified trtc appid of the digital human platform is used, limited to the debugging stage and cannot be used in the product
TrtcAppId	string	No	The trtc appid (required when using an external trtc appid).
TrtcRoomId	int	No	The trtc room ID (The room ID will be assigned by the cloud if it is specified.).
TrtcUserSig	string	No	The trtc AI Digital Human user signature (required when using ar trtc appid).
TrtcPrivateMapKey	string	No	The trtc AI Digital Human user permission key (required when us external trtc appid).

CssCustomPushUrl	string	Yes	Custom Cloud Streaming Services (CSS) streaming URL. The st protocol is fixed to use rtmp. By using rtmp to push the stream to various pull protocols supported by CSS can be used for playbac
			The required format for the streaming URL is:
			`rtmp://xyz.com/cssAppName/streamid?txSecret={0}&txTime={1}`
			For the calculation method of txSecret and txTime, see
			`https://cloud.tencent.com/document/product/26`

ExtraInfo

Name	Type	Required	Description
AlphaChannelEnable	bool	No	Enable original video concatenation alpha to output video stream. **Notes:** Assuming the original output width is iW, the actual video stream width is: iW + pW + iW.

			pW is the middle padding width: (16 - iW%16) % 16 alpha video width starting point: iW + pW alpha video width: iW

Response Parameter

Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
SessionId	String	Yes	Unique identifier for the session
SessionStatus	int	Yes	Status: 1: In progress (ready), the cache hit directly provides the playback URL. 3: Preparing (not ready), the cache missed requires waiting for model to load. Use the Querying the Session Status API to poll the session status until the stream status changes to 1. This usually takes no more than 2 minutes.
PlayStreamAddr	String	No	The playback URL, in the format: `rtmp://liveplay.ivh.qq.com/live/m789` This field will not be returned if a custom CSS streaming URL is specified through the CssCustomPushUrl parameter.

Request Sample

```
{
    "Header": {

    },
    "Payload": {
        "ReqId": "f2612aa810014e8997f95bda97917268",
        "VirtualmanProjectId": "1b2c9f1c268a4edfbf6c6274da31bc5d",
        "UserId": "f2612aa810014e8997f95bda97917268",
        "Protocol": "rtmp",
        "DriverType": 1
```

```
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m789",
        "SessionStatus": 1,
        "PlayStreamAddr": "rtmp://live.qq.com/live/m789"
    }
}
```

Using the Personal Asset Image to Create a Stream

Last updated：2025-01-17 16:24:37

It is used to create a new video stream and obtain the playback URL for the digital human video.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/sessionmanager/sessionmanagerservice/createsessionbyasset
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
SessionId	String	No	A unique identifier for the session. The SaaS environment Private Environment allows input.
AssetVirtualmanKey	String	Yes	Personal asset image ID. It is obtained from the Asset Ma as shown in the image:
UserId	String	Yes	The unique identifier for the user, maintained by the caller UserIdentifier will cause the previous stream with that Use
Protocol	String	Yes	Currently supported parameters: rtmp, trtc and webrtc
DriverType	int	Yes	The driving method for the digital human.

			1: text-only driving; 3: audio driving (original voice), which supports both audi
ProtocolOption	ProtocolOption	No	Custom parameters for the protocol; ignore trtc if it is not u
SpeechParam	SpeechParam	No	Parameters related to timbres.

ProtocolOption

Name	Type	Required	Description
TrtcUseExternalApp	boolean	No	It indicates whether to use an external trtc. If you do not use it, th trtc appid of the digital human platform will be used. **Notes:** When the unified trtc appid of the digital human platform is used, limited to the debugging stage and cannot be used in the product
TrtcAppId	string	No	The trtc appid (required when using an external trtc appid).
TrtcRoomId	int	No	The trtc room ID (The room ID will be assigned by the cloud if it is specified.).
TrtcUserSig	string	No	The trtc AI Digital Human user signature (required when using ar trtc appid).
TrtcPrivateMapKey	string	No	The trtc AI Digital Human user permission key (required when us external trtc appid).

CssCustomPushUrl	string	No	Custom Cloud Streaming Services (CSS) streaming URL. The st protocol is fixed to use rtmp. By using rtmp to push the stream to various pull protocols supported by CSS can be used for playbac
			The required format for the streaming URL is:
			`rtmp://xyz.com/cssAppName/streamid?txSecret={0}&txTime={1}`
			For the calculation method of txSecret and txTime, see
			`https://cloud.tencent.com/document/product/26`

SpeechParam

Name	Type	Required	Description
Speed	float	No	Speech speed, with a range of [50, 200]. A value of 50 indicates 50% of the default speed. If the speed is not

			specified, it is recommended to use the default value of 100.
TimbreKey	string	No	Timbre key.
Volume	int	No	Volume level, with a range of [-10, 10]. A value of -10 indicates a decrease of 10 dB relative to the default volume. If the volume is not specified, it is recommended to use the default value of 0.
TimbreLanguage	string	No	Timbre languages, optional languages can be found in the Personal Asset Management API Paginated Query Timbre List API. For multilingual timbres, the corresponding language must be selected during synthesis.

Response Parameter

Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
SessionId	String	Yes	Unique identifier for the session
SessionStatus	int	Yes	Status: 1: In progress (ready), the cache hit directly provides the playback URL. 3: Preparing (not ready), the cache missed requires waiting for model to load. Use the Querying the Session Status API to poll the session status until the stream status changes to 1. This usually takes no more than 2 minutes.
PlayStreamAddr	String	No	The playback URL, in the format: `rtmp://liveplay.ivh.qq.com/live/m789` This field will not be returned if a custom CSS streaming URL is specified through the CssCustomPushUrl parameter.

Request Sample

```
{
    "Header": {
```

```
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "AssetVirtualmanKey": "d7aa08da33dd4a662ad5be508c5b77cf",
        "DriverType": 1,
        "UserId": "henry",
        "Protocol": "rtmp"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m789",
        "SessionStatus": 1,
        "PlayStreamAddr": "rtmp://live.qq.com/live/m789"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Querying the Session Status

Last updated：2024-07-19 10:02:16

It is used to query the current status of a specified session.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/statsession

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Required	Type	Description
ReqId	Yes	String	Unique identifier for each request, a 32-character UUID.
SessionId	Yes	String	Unique identifier for the session.

Response Parameter

Parameter name	Required	Type	Description
ReqId	Yes	String	Unique identifier for a single request.
SessionStatus	Yes	int	Stream status.1: in progress; 2: closed; 3: preparing; 4: stream creation failed
PlayStreamAddr	No	string	Streaming playback address
SpeakStatus	No	string	Status of the digital human. Initial: initial state. WaitingTextStart: waiting for text broadcast to start. TextStart: text broadcasting. WaitingTextOver: waiting for text broadcast to finish. TextOver: text broadcast finished. WaitingAudioStart: waiting for audio broadcast to start. AudioStart: audio broadcast started. WaitingAudioOver: waiting for audio broadcast to finish. AudioOver: audio broadcast finished. Error: drive error

©2013-2025 Tencent Cloud International Pte. Ltd.

			(same as TextOver and AudioOver, indicating the end state of a drive). It only denotes the failure of the most recent drive command and does not affect the ability to continue sending drive commands.
IsSessionStarted	Yes	bool	It indicates whether the session has started. Drive instructions can only be sent when the session is active. **Note:** When false is returned, you need to call the Start Session API to initiate.
ErrorCode	Yes	int	The error code corresponding to the current status; 0 indicates normal operation. See Section 7.2 for the list of error codes.
ErrorMessage	No	string	The error message corresponding to the current status.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionStatus": 1,
        "PlayStreamAddr": "rtmp://live.qq.com/live/m789",
        "SpeakStatus": "TextOver",
        "IsSessionStarted": true,
```

```
        "ErrorCode": 0,
        "ErrorMessage": ""
    }
}
```

Starting Session

Last updated：2024-07-19 09:59:33

After the stream is ready, this API must be called to proceed with driving.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/startsession

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
SessionId	String	Yes	Unique identifier for the session.

Response Parameter

Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "SessionId": "m123",
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf"
    }
}
```

Closing Live Session

Last updated：2024-07-19 09:59:49

Close the live session, stop the digital human streaming, and release concurrency.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/closesession

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
SessionId	String	Yes	Unique identifier for the session.

Response Parameter

Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "SessionId": "m123adfafvbadsafd",
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "m123adfafvbadsafd",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

Querying the Conversation List under the Digital Human Project

Last updated：2024-07-19 10:00:06

It is used to query all ongoing conversation lists under the digital human project.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/listsessionofprojectid

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
VirtualmanProjectId	String	Yes	Digital Human Project ID.

Response Parameter

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
Sessions	Session[]	Yes	Session list array

Session

Name	Type	Required	Description
UserId	String	Yes	User's unique identifier
SessionId	String	Yes	Unique identifier for the session.

SessionStatus	int	Yes	Stream status.1: in progress; 2: closed; 3: preparing; 4: stream creation failed
PlayStreamAddr	String	No	Stream playback address, returned upon successful stream creation
DriverType	String	Yes	Digital human type. 1: text-driven; 3: voice-driven (original sound).
IsSessionStarted	bool	Yes	It indicates whether the session has started. Drive instructions can only be sent when the session is active.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "Sessions": [
            {
                "UserId": "abc",
                "SessionId": "m123",
                "SessionStatus": 1,
                "PlayStreamAddr": "rtmp://live.qq.com/live/m789",
                "DriverType": 1,
                "IsSessionStarted": true
            }
```

```
            ]
        }
    }
```

Querying the Conversion List under the Personal Asset Image

Last updated：2024-07-19 10:01:26

It is used to query all ongoing conversation lists under the personal asset image.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/listsessionofassetvk

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
AssetVirtualmanKey	String	Yes	Personal Asset Image ID.

Response Parameter

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
Sessions	Session[]	Yes	Session list array

Session

Name	Type	Required	Description
UserId	String	Yes	User's unique identifier
SessionId	String	Yes	Unique identifier for the session.

SessionStatus	int	Yes	Stream status.1: in progress; 2: closed; 3: preparing; 4: stream creation failed
PlayStreamAddr	String	No	Stream playback address, returned upon successful stream creation
DriverType	String	Yes	Digital Human type. 1: text-driven; 3: voice-driven (original sound).
IsSessionStarted	bool	Yes	It indicates whether the session has started. Drive instructions can only be sent when the session is active.

Request Sample

```
{
    "Header": {

    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "AssetVirtualmanKey": "d7aa08da33dd4a662ad5be508c5b77cf"
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "Sessions": [
            {
                "UserId": "abc",
                "SessionId": "m123",
                "SessionStatus": 1,
                "PlayStreamAddr": "rtmp://live.qq.com/live/m789",
```

```
                "DriverType": 1,
                "IsSessionStarted": true
            }
        ]
    }
}
```

Querying the Session List under UIN

Last updated：2024-07-19 10:01:45

It is used to query all ongoing session information for a specific UIN account.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/listsessionofuin

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.

Response Parameter

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
Sessions	Session[]	Yes	Session list array

Session

Name	Type	Required	Description
UserId	String	Yes	User's unique identifier
SessionId	String	Yes	Unique identifier for the session.
SessionStatus	int	Yes	Stream status.1: in progress; 2: closed; 3: preparing; 4: stream creation failed
PlayStreamAddr	String	No	Stream playback address, returned upon successful stream creation

DriverType	String	Yes	Digital human type. 1: text-driven; 3: voice-driven (original sound).
IsSessionStarted	bool	Yes	It indicates whether the session has started. Drive instructions can only be sent when the session is active.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "Sessions": [
            {
                "UserId": "abc",
                "SessionId": "m123",
                "SessionStatus": 1,
                "PlayStreamAddr": "rtmp://live.qq.com/live/m789",
                "DriverType": 1,
                "IsSessionStarted": true
            }
        ]
    }
}
```

Instruction-driven
Instruction Sending Requirements

Last updated：2024-07-19 10:03:07

Note:

Based on a persistent connection, instructions are sent to drive the AI digital human.

Persistent connection instructions are divided into sending text instructions, sending audio instructions, and sending heartbeat instructions.

Sending a text instruction involves sending broadcast text to the cloud, which synthesizes speech to drive the AI digital human to speak;

Sending an audio instruction involves sending voice stream shards to the cloud, which then uses the original sound or voice changer to drive the AI digital human to speak;

Sending a heartbeat instruction is to maintain a persistent connection and prevent session disconnection. If the client does not send a heartbeat and there is no valid data exchanged, the cloud will automatically disconnect the persistent connection after 3 minutes and close the session stream after 10 minutes.

The relationship between the DriverType parameter and the instruction types when creating a new stream is as follows:

DriverType 1: Text-driven; supporting sending text instructions (SEND_TEXT) and heartbeat instructions (SEND_HEARTBEAT).

DriverType 3: Audio-driven; supporting sending text instructions (SEND_TEXT), audio instructions (SEND_AUDIO), and heartbeat instructions (SEND_HEARTBEAT).

1. When a text is being broadcast, if you want to send audio, you need to send an interrupt with an empty text instruction until you receive a TextOver event, after which you can send the audio.

2. When the text is being broadcast and you want to broadcast new text, directly send the new text instruction. The current text broadcast will be interrupted, and upon completion, you will receive a TextOver event. The broadcast of the new text will start with a TextStart event and end with a TextOver event.

3. When an audio-driven session is in progress, if you want to interrupt the audio to send a text instruction, you need to send an audio instruction with IsFinal set to true to end the audio session. Only after receiving an AudioOver event can you send text or audio.

©2013-2025 Tencent Cloud International Pte. Ltd.

Creating a Long Connection Channel

Last updated：2024-07-19 10:11:28

Establish a WebSocket long connection channel to send drive instructions upstream and receive downstream messages.

Calling Protocol

WebSocket + JSON
WSS /v2/ws/ivh/interactdriver/interactdriverservice/commandchannel
Header Content-Type: application/json;charset=utf-8

Connection Establishment Method

websocket url: see Digital Human aPaas API Calling Method for organizing the URL.

Note:

It is particularly important to note that the requestid parameter should be filled with the value of the session ID (SessionId). For example:

```
wss://domin/v2/ws/ivh/interactdriver/interactdriverservice/commandchannel?appkey=xxx&requestid=m123&timestamp=xxx&signature=xxx
```

Websocket may disconnect due to network fluctuations, cloud service updates, etc. When a disconnection is detected, as long as the session status has not been closed, re-establishment is required.

Please call theStart Session API before establishing a WebSocket long connection; otherwise, the connection will fail. Only one long connection channel is allowed per session.

©2013-2025 Tencent Cloud International Pte. Ltd.

Text-driven Instructions

Last updated：2024-07-19 10:06:40

After you Create Long Connection Channel, you can use a websocket persistent connection to send the text to drive the digital human.

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single drive session.
SessionId	String	Yes	Unique identifier for the session.
Command	String	Yes	SEND_TEXT: Send text.
Data	Data	Yes	Data Object

Data

(Send text.)

Name	Type	Required	Description
Text	String	No	The text content to be broadcasted, default value: maximum length of 4000 bytes.
Interrupt	Boolean	No	Forced interruption flag. The default value is false. If forced interruption is not used and the text is marked as non-interruptible on the digital human platform, you will receive feedback indicating that interruption is not possible. If forced interruption is used, the system will ignore the non-interruptible setting on the digital human platform and execute the forced interruption directly.
ChatCommand	String	No	Dialogue instruction type, effective when the digital human project is bound to a customer service chatbot. NotUseChat: The Q&A knowledge library configured for the customer service bot is not effective, and the digital human broadcasts the content sent. Pass empty or do not pass: The Q&A knowledge library configured for the customer service bot

©2013-2025 Tencent Cloud International Pte. Ltd.

			becomes effective, and the digital human broadcasts the answers from the Q&A knowledge library.
ChatRoundId	String	No	Conversation round ID, recommended to use UUID. Default value: "". For digital human projects using Tencent's industry large model conversation service, this field needs to be provided to distinguish between multiple rounds of dialogue.
CloudAiVisitorBizId	String (64)	No	Visitor ID for Tencent's large model knowledge engine.

Warning:

The interval between sending texts must be greater than 1 second.

Parameter example

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_TEXT",
        "Data": {
            "Text": "Hello, I am an artificially synthesized digital human",
            "ChatCommand": "NotUseChat"//Using the text-driven stream built for the
        }
    }
}
```

Voice-driven Instructions

Last updated：2024-07-19 10:08:06

After you Create Long Connection Channel, you can use a websocket persistent connection to send audio to drive the digital human.

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	A unique identifier for a single drive. Each segment of audio is assigned a UUID value.
SessionId	String	Yes	Unique identifier for the session.
Command	String	Yes	SEND_AUDIO; send the audio.
Data	Data	Yes	Data Object

Data

Name	Type	Required	Description
Audio	string	Yes	The byte array of the original audio data, encoded into a string via Base64. Only supports: format-PCM, sampling rate-16kHz, sampling bit depth-16bits, audio track-mono.
Seq	int	Yes	Audio packet sequence number, which must start from 1.
IsFinal	bool	No	The default value is false.

Note:

1. If the data is being sent in real-time from a microphone, it can be sent every 160 ms (5120B) without any waiting interval. If the data is being sent from an offline audio file, the packet size should be 160 ms (5120B) with a 120 ms interval between packets.
2. The size of the last packet should be based on the actual remaining data (must be less than 160 ms).
3. After all data packets have been sent, an empty data packet with IsFinal=true (with the Audio field left empty) must be sent to signal the end of the audio session and return the Digital Human to a silent state.
4. The real-time rate of sending audio must be between [0.75, 1]. A rate lower than 0.75 will trigger throttling, while a rate higher than 1 will cause video stuttering. For example, for a 160 ms audio packet size, the sending interval must

©2013-2025 Tencent Cloud International Pte. Ltd.

not be less than 120 ms or more than 160 ms.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_AUDIO",
        "Data": {
            "Audio": "The value of the audio binary data encoded in Base64",
            "Seq": 0,
            "IsFinal": false
        }
    }
}
```

Streaming Text-driven Instructions

Last updated：2024-07-19 10:13:04

After you Create a Long Connection Channel, you can use a WebSocket persistent connection to send streaming text to drive the digital human.

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	A unique identifier for a single drive. Each segment of streaming text is assigned a UUID value.
SessionId	String	Yes	Unique identifier for the session.
Command	String	Yes	SEND_STREAMTEXT; send streaming text.
Data	Data	Yes	Data Object

Data

Name	Type	Required	Description
Text	string	Yes	Streaming text content only requires incremental text transmission, with each fragment packet string limited to 1000 bytes in length.
Seq	int	Yes	Sequence number of the streaming text fragment packet. The sequence must start from 1.
IsFinal	bool	No	The default value is false.
Interrupt	bool	No	Forced interruption marker. The default value is false. Passing in true and an empty Text means interrupting the streaming text.
SmartActionEnabled	bool	No	Whether smart actions are enabled; the default value is false. When it is set to true and the input text or enhanced text lacks action tags, smart actions will be generated.
IsSentence	bool	No	Whether it is in clause mode; the default value is false.

©2013-2025 Tencent Cloud International Pte. Ltd.

			When it is set to true the server will not rearrange sentences.
IsInsertSentence	bool	No	Indicating whether it is an inserted clause. The default value is false. When it is set to true and in clause mode, it indicates that the current segment needs to be inserted.

Request Sample

```
//Normal streaming text fragment packet{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_STREAMTEXT",
        "Data": {
            "Text": "Streaming Text Content",
            "Seq": 1,
            "IsFinal": false
        }
    }
}//Streaming text final packet{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_STREAMTEXT",
        "Data": {
            "Text": "The streaming text has ended.",
            "Seq": 59,
            "IsFinal": true
        }
    }
}//Interrupt the currently broadcasting streaming text or streaming text in clause
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_STREAMTEXT",
        "Data": {
            "Text": "",
```

```
            "Seq": 59,
            "IsFinal": true,
            "Interrupt": true
        }
    }
}//Normal streaming text clause mode fragment packet{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_STREAMTEXT",
        "Data": {
            "Text": "Streaming text clause mode content",
            "Seq": 1,
            "IsFinal": false,
            //Streaming text in clause mode does not require sending a final packet
            "IsInsertSentence": false//Set this value to true if it is an inserted
        }
    }
}
```

Long Link Downstream Message

Last updated：2024-07-19 10:13:14

After you Creating a Long Connection Channel, the digital human server will return the real-time digital human driving status information to the client.

Name	Type	Required	Description
Type	int	Yes	Downstream message types. Other return types for Avatar can be ig 3: broadcast status. 4: content returned by the large model. 9: driving failed.
SessionId	String	Yes	Unique identifier for the video stream session
ReqId	String	Yes	Unique identifier for a single drive session
Seq	int	Yes	Unique sequence number of the streaming clause
SpeakStatus	String	No	Initial: initial status WaitingTextStart: waiting for text broadcast to start TextStart: text broadcast in progress WaitingTextOver: waiting for text broadcast to end TextOver: Text broadcast ended. WaitingAudioStart: waiting for audio broadcast to start AudioStart: Audio broadcast started. WaitingAudioOver: waiting for audio broadcast to end AudioOver: Audio broadcast ended. Error: Driving failed (similar to TextOver and AudioOver, indicating th session). This only indicates the most recent drive failure and does n continue sending drive commands. SentenceNext: In streaming clause mode, this status is returned, indi that it can send the next clause upon receiving this status. SentenceStart: In streaming clause mode, this status is returned, ind that the current clause playback has started upon receiving this statu SentenceOver: In streaming clause mode, this status is returned, ind that the current clause playback has ended upon receiving this status
FinalType	int	No	When SpeakStatus returns AudioOver, it has a value indicating the s packet that marks the end of audio playback： 1 - Client input; 2 - Server-generated packet due to timeout
Text	String	No	This field is present when Type is 2 or 4, indicating a question promp

©2013-2025 Tencent Cloud International Pte. Ltd.

TextPro	String	No	This field is present when the Type is 2 or 4, containing the broadcas tags.
TextDisplay	String	No	This field is present when Type is 2 or 4, containing the broadcast te client.
ContentType	String	No	This field is present when the Type is 4, used to distinguish the conte the large model: Image: Image Hyperlink: Hyperlink OrderedList: Ordered list UnOrderedList: Unordered list List: Table
TtsSupport	bool	No	This field is present when the Type is 4, indicating whether the conte large model needs to be broadcast via tts.
Final	bool	No	This field is present when the Type is 4, indicating whether the conte large model is the final clause.
IsHighLight	bool	No	This field is present when the Type is 4, indicating whether the conte large model needs to be highlighted on the client side.
Uninterrupt	bool	No	This field is present when the Type is 2 or 4, indicating whether the c sentence can be interrupted: true: Cannot be interrupted false: Can be interrupted
Muted	bool	No	This field is present when the Type is 2 or 4, indicating whether the c has the microphone muted.
InteractionType	String	No	This field is present when the Type is 2, and can be used for platform Common types include popup, image, etc.
InteractionContent	String	No	This field is present when the Type is 2, used for special messages s images, and other non-text content.
ErrorCode	int	Yes	Error code, with a value of 0 indicating normal operation. A non-zero error, such as an invalid request body, incorrect timing of drive comm request rate. See Section 7 for the error code list.
ErrorMessage	String	No	Error Description
CloudAiExtra	String	No	Extended fields returned by Tencent's LLM Knowledge Engine. Resp {"record_id":"b24e6505-be78-4ce9-ad9e-7075dd227994","reference [{"id":"17804349457144217 60","type":2,"url":"https://oaqbot.qidian.q id=178022705279008768 0","name":"Client meeting","doc_id":"1129 For the like/dislike API, see Evaluation Message.

			For specific reference to the source field, see MsgRecordReference.

Note:

When websocket reconnects, the cloud will proactively push the playback status triggered by the last three drive requests to prevent the client from losing corresponding drive feedback events due to network instability during the reconnection process.

Heartbeat Instruction

Last updated：2024-07-19 10:13:22

If the cloud does not receive any drive instructions for more than 10 minutes, the connection will be automatically disconnected, and concurrency will be released. After you Create Long Connection Channel, the session can be extended by sending a heartbeat instruction.

Note:

The sending interval should be greater than 30 seconds and less than 10 minutes.

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for each request, a 32-character UUID.
SessionId	String	Yes	Unique identifier for the session.
Command	String	Yes	SEND_HEARTBEAT: Send a heartbeat.
Data	Data	Yes	Data Object

Data

Name	Type	Required	Description
Text	string	Yes	Use the fixed value PING

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "Command": "SEND_HEARTBEAT",
        "Data": {
            "Text": "PING",
        }
```

```
        }
    }
```

Other APIs
Real-time Update of Digital Human Visual Parameters

Last updated：2024-07-19 10:13:42

It is used for real-time updating of the character's size, position, and background image in the video stream.

Note:

This API is no longer recommended for use, and there is no further iteration of new features. For adjusting the background and the size of the figure, it is recommended to use the following method: set up green screen background streaming and use streaming software to chroma key and overlay foreground and background, and adjust character size.

Note:

Request frequency limit for this API: One request every 5 seconds per individual session.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/updatesessionconfig

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
SessionId	String	Yes	Session ID.
SpaceParam	SpaceParam	No	Spatial information. This parameter is supported by Avatar.
ImageParam	ImageParam	No	Avatar information. This parameter is supported by 3D avatars.

SpaceParam

Name	Type	Required	Description
BackgroundUrl	String	No	Background URL address (less than 1 MB in size, preferably aligned with output resolution to avoid stretching and filling) Requirements: jpg, jpeg, and png (other formats not supported). If it is not provided, the previous background image will be used.
ImagePosition	float	Yes	The X position of the character in the background image, with the character's image centered on the axis where X is 0. The range of values is linear from -0.5 to 0.5. The character's position cannot exceed the boundaries of the background image.
ImagePositionVertical	float	Yes	The Y position of the character in the background image, with the character's image aligned to the bottom (Y=0). The range of values is linear from -0.5 to 0.25, allowing the character to extend downward beyond the boundaries of the background image.
ImageZoom	float	Yes	The zoom ratio of a character relative to the original material image, calculated based on the size after zooming and the original character material image. The calculation formula is: width of the zoomed character image / (background width of the preview area / target resolution width) / width of the original character material image. The recommended value range is [0.25, 2].

ImageParam

Name	Type	Required	Description
ImageVerticalOffset	float	No	The vertical rotation angle of the upper body of a 3D character
ImageHorizontalOffset	float	No	The horizontal rotation angle of the upper body of a 3D character
ImageExtraParam	float	No	Color change parameters for the 3D character. Specific parameters can be found on the interactive digital human platform.

Response Parameter

Name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "m123adfafvbadsafd",
        "ImageParam": {
            "ImageVerticalOffset": 10,
            "ImageHorizontalOffset": 10,
            "ImageExtraParam": "{\\"JacketColor\\":{\\"colorValue\\":10491928},\\"D
        },
        "SpaceParam": {
            "BackgroundUrl": "https://virtualhuman-cos-prod-1251316161.cos.ap-nanji
            "ImagePosition": -0.01,
            "ImagePositionVertical": -0.01,
            "ImageZoom": 1
        }
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
    }
}
```

Text-driven Instructions

Last updated：2024-07-19 10:14:37

Calling Protocol

HTTPS + JSON

POST /v2/ivh/interactdriver/interactdriverservice/command

Header Content-Type: application/json;charset=utf-8

Note:

The interval between sending texts must be greater than 1 second.

Request Parameters

Parameter name	Type	Required	Description
ReqId	String	Yes	Unique identifier for a single request.
SessionId	String	Yes	Unique identifier for the session.
Command	String	Yes	SEND_TEXT: Send text.
Data	Data	Yes	Data Object

Data

(Send text.)

Name	Type	Required	Description
Text	string	No	The text content to be broadcasted, default value: "" The string with a maximum length of 4000.
Interrupt	boolean	No	Forced interruption flag. The default value is false. If forced interruption is not used and the text is marked as non-interruptible on the digital human platform, you will receive feedback indicating that interruption is not possible. If forced interruption is used, the system will ignore the non-interruptible setting on the digital human platform and execute the forced interruption directly.

©2013-2025 Tencent Cloud International Pte. Ltd.

ChatCommand	string	No	Dialogue instruction type, effective when the digital human project is bound to a customer service chat bot. NotUseChat: The Q&A knowledge library configured for the customer service bot is not effective, and the digital human broadcasts the content sent. Pass empty or do not pass: The Q&A knowledge library configured for the customer service bot becomes effective, and the digital human broadcasts the answers from the Q&A knowledge library.

Response Parameter

There is no need to pay attention to the returned content, just focus on the returned error code.

Request Sample

```
//Send the text message.
{
    "Header": {},
    "Payload": {
        "SessionId": "m123",
        "Command": "SEND_TEXT",
        "Data": {
            "Text": "Hello"
        }
    }
}
//After the customer service chat is bound, don't use the chat; broadcast the conte
{
    "Header": {},
    "Payload": {
        "SessionId": "m123",
        "Command": "SEND_TEXT",
        "Data": {
            "Text": "Hello",
            "ChatCommand": "NotUseChat"//Using the text-driven stream built for the
        }
    }
}

//Force interruption
{
```

```
    "Header": {},
    "Payload": {
        "SessionId": "m123",
        "Command": "SEND_TEXT",
        "Data": {
            "Interrupt": true
        }
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "m123adfafvbadsafd",
    }
}
```

Obtaining the Key to the TRTC Room

Last updated：2024-07-19 10:14:53

When a live session is created and the TRTC protocol is selected, this API is used by users other than the Digital Human to obtain the key for entering the room.

Note:

This only applies to situations where the internal TRTC account of the digital human is used.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/sessionmanager/sessionmanagerservice/gettrtcsign

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter name	Required	Type	Description
ReqId	Yes	String	Unique identifier for a single request.
SessionId	Yes	string	Session ID.
RoomId	Yes	string	The room ID, obtained through API 4.1 and parsed from the returned addr address.
UserId	Yes	string	User ID for entering the room.

Response Parameter

Parameter name	Required	Type	Description
ReqId	Yes	String	Unique identifier for a single request.
UserSig	Yes	string	Room Key.
PrivateMapKey	Yes	string	Permission Key.

©2013-2025 Tencent Cloud International Pte. Ltd.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "SessionId": "12123132",
        "RoomId": "1200111",
        "UserId": "avcvdafasfds",
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123",
    },
    "Payload": {
        "ReqId": "d7aa08da33dd4a662ad5be508c5b77cf",
        "UserSig": "vafdasfda",
        "PrivateMapKey": "fdasfdafdasfdasfd"
    }
}
```

©2013-2025 Tencent Cloud International Pte. Ltd.

Error Code Description

Last updated：2024-07-19 10:15:31

Feature Description

When the code value in the returned result is not 0, it indicates that the API call has failed. For example:

```
{
    "Header": {
        "Code": 100000,
        "Message": "Internal system error",
        "RequestID": "123",
    }
}
```

Error Code List

Code	Meaning	Measures
-101000	Internal system error	Retry or contact customer service to locate the error.
-101001	The computing resources for the digital human service are insufficient.	Retry or contact customer service to locate the error.
-101002	Image downloading failed.	Retry or contact customer service to locate the error.
-111001	Updating stream parameters failed.	Retry or contact customer service to locate the error.
-111002	Connection abnormally disconnected.	Retry or contact customer service to locate the error.
-111003	An error occurred during the model loading process of the digital human.	Retry or contact customer service to locate the error.
-111004	There was an internal stream creation timeout in the digital human service.	Retry or contact customer service to locate the error.

©2013-2025 Tencent Cloud International Pte. Ltd.

-111005	An error occurred when the digital human with text or audio was driven.	Please read the description of the abnormal disconnection. If it is a system error, contact the customer service to locate the error.
100001	The request parameters were incorrect.	For errors related to field format, length, type, or non-null constraints, refer to the protocol documentation and read the field transmission requirements.
100005	Unauthorized operation	Please request the relevant API or resource according to the authorized content.
100008	Exceeding the concurrency limit	Please check the used concurrency number and the upper limit of the concurrency number.
100012	Requests are too frequent.	Please send requests according to the frequency specified in the protocol.
100014	The concurrency quota is insufficient.	Please check if the concurrency quota is sufficient or if it is within the valid period.
100015	The digital human avatar has expired.	Please check the validity period.
110013	The session has been closed.	Please check if you are driving a closed stream. If the stream was closed abnormally, use statsession to check the code and message to locate the cause of the abnormal closure.
110014	The session is not ready.	Please confirm that the stream creation is complete with status as 1 and that startsession has been used to start the session.
110015	The request was not processed due to the incorrect timing of sending the drive command.	Please send the command according to the requirements specified in 5.1.
110016	A drive command was sent to an inactive session.	Please confirm that the stream creation is complete with status as 1 and that startsession has been used to start the session.
110018	The session does not exist.	Please check if the session has been closed or if it was not created.
110021	TRTC service is overdue, or its available time is insufficient.	Please check if the TRTC service is overdue.
110022	The stream session created with the same userid has been closed.	Please check the userid transmitted during stream creation.

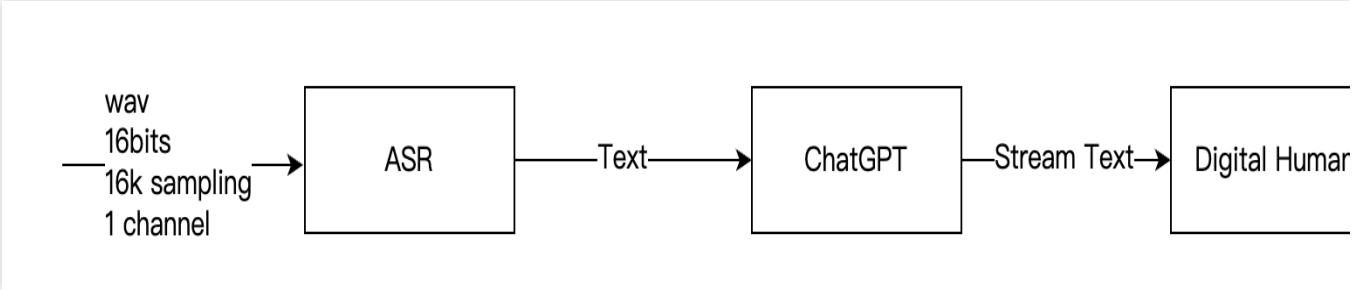
©2013-2025 Tencent Cloud International Pte. Ltd.

Best Practices demo

Last updated：2024-08-14 09:20:05

ASR/ChatGPT/digital human integration demo

This demo shows how to interact with the digital human through ASR/ChatGPT.



demo Content	Language	Latest Version File Name	demo Download Link
ASR/ChatGPT/Digital Human Integration demo	python	asr_chatgpt_vh_demo_20240812.zip	Click to download

Note:

For details about the demo operation method, read the README.md file in the demo package carefully.

Personal Asset Management API Documentation
Overview

Last updated：2024-07-19 10:15:56

This document mainly describes the customer asset management API. Customers can use this set of APIs to query their currently activated digital human assets, including **Image Assets** and **Service Assets**. Service assets include broadcast hourly packages, broadcast concurrency, and interactive concurrency.

Note:

Currently, only the Avatar image asset is supported for querying. For 2D premium and 3D images, see the Customer Resource Query Anchor API.

API Calling Methods

For details, see Digital Human aPaaS API Calling Methods.

Digital Human aPaaS API Calling Methods

Last updated：2024-07-19 10:17:06

For details, see Digital Human aPaaS API Calling Methods.

©2013-2025 Tencent Cloud International Pte. Ltd.

Querying for Avatar List by Pagination API

Last updated：2024-07-19 10:17:21

Query the Avatar list through pagination parameters.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/crmserver/customerassetservice/describesmallsampleimage

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
PageIndex	int	Yes	Current page
PageSize	int	Yes	Page size (maximum of 100)
AnchorCodes	Array of string	No	Filter condition by AnchorCode

Response Parameter

Parameters	Type	Mandatory	Description
Virtualmans	Array of [SmallSampleImage]	Yes	Resources of the digital human within customers' permissions
Total	int	Yes	Totals

SmallSampleImage

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Digital human virtualmanKey; a unique identifier for the avatar.
AnchorName	string	Yes	Anchor Name

AnchorCode	string	Yes	Anchor Code
ClothesName	string	Yes	Digital human clothing
PoseName	string	Yes	Digital human pose
Resolution	string	Yes	Digital human resolution
HeaderImage	string	Yes	Digital human avatar image URL
PoseImage	string	Yes	Digital human pose image URL
ClothesImage	string	Yes	Digital human clothing image URL
SupportDriverTypes	Array of [string]	Yes	Supported drive types of the digital human 1. Text: Text-driven 2. OriginalVoice: Original voice audio-driven 3. ModulatedVoice: Modulated voice audio-driven
ExpireDate	string	Yes	Validity Period
VideoDuration	int	Yes	Total accumulated duration for calling the digital human, in milliseconds.
VideoNum	int	Yes	Total number of videos for the digital human
OriginZoom	float	Yes	Initial zoom factor for the digital human
OriginZoomStr	String type floating point number	Yes	Initial scaling factor of the digital human, accurate to three decimal places
ReferenceVideoSegmentUrl	string	Yes	URL of the selected video clip for the Avatar.

Example

Input parameters:

```
{
    "Header": {},
    "Payload": {
        "pageIndex": 1,
```

```
        "PageSize": 10
    }
}
```

Output response:

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Virtualmans": [
            {
                "VirtualmanKey": "6d266ff720f04df39b8e0a17553382d1",
                "AnchorName": "Teacher Wuji",
                "AnchorCode": "wuji_teacher_v3",
                "ClothesName": "polo shirt",
                "PoseName": "Standing",
                "Resolution": "1920x1080",
                "HeaderImage": "url",
                "PoseImage": "url",
                "ClothesImage": "url",
                "SupportDriverTypes": [
                    "Text",
                    "OriginalVoice"
                ],
                "ExpireDate": "2023-04-01 00:00:00",
                "VideoDuration": 50000,
                "VideoNum": 3,
                "OriginZoom": 0.50
            }
        ],
        "Total": 22
    }
}
```

Querying Supported Timbres for Avatars (to be Deprecated)

Last updated：2024-07-19 10:17:38

Query the supported timbres based on the avatar's VirtualmanKey.

Note:

The current digital human avatar and timbre have been decoupled, meaning that all avatars can use the Paginated Query Timbre List API to generate videos. It is not recommended to use this API.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/crmserver/customerassetservice/getimagetimbre

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	The unique identifier of the avatar returned by the Paginated Query Avatar List API.

Response Parameter

Parameters	Type	Mandatory	Description
Timbres	Array of [Timbre]	Yes	Supported Timbre List

Timbre

Parameters	Type	Mandatory	Description
TimbreKey	string	Yes	Digital human timbre code
TimbreName	string	Yes	Digital human timbre name

TimbreSample	string	Yes	Digital human timbre demo sample URL
TimbreDesc	string	Yes	Digital human timbre description

Example

Input parameters:

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "963a2a7d4c734a6ca2984835d9eae765"
    }
}
```

Output response:

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "Timbres": [
            {
                "TimbreKey": "female_1",
                "TimbreName": "female voice 1",
                "TimbreSample": "url",
                "TimbreDesc": "Natural and graceful",
            }
        ]
    }
}
```

Querying Customer Service Asset Information

Last updated：2024-07-19 10:17:54

Query customer service assets to find information about the customer's current broadcast hourly package, broadcast concurrency, interaction concurrency, customizable avatar quantities, and customizable voice clone quantities.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/crmserver/customerassetservice/getserviceasset
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
ServiceType	string	No	Service asset type, see Appendix 1 - Service Asset Types for specific types. If not specified, all types will be queried by default.
PageIndex	int	Yes	Current page
PageSize	int	Yes	Page size (maximum of 100)

Response Parameter

Parameters	Type	Mandatory	Description
ServiceAssetInfos	Array of [ServiceAssetInfo]	Yes	Service Asset Information
Total	int	Yes	Totals

ServiceAssetInfo

Parameters	Type
ServiceType	string

ExpireDate	string
InteractConcurrencyInfoDetail	[InteractConcurrencyInfoDetail]
BroadcastHourInfoDetail	[BroadcastHourInfoDetail]
BroadcastConcurrencyInfoDetail	[BroadcastConcurrencyInfoDetail]
ImageAssetInfoDetail	[ImageAssetInfoDetail]
TimbreAssetInfoDetail	[TimbreAssetInfoDetail]
RealMan3DInteractConcurrencyInfoDetail	[RealMan3DInteractConcurrencyInfoDetail]
RealMan3DBroadcastHourInfoDetail	[RealMan3DBroadcastHourInfoDetail]
RealMan3DBroadcastConcurrencyInfoDetail	[RealMan3DBroadcastConcurrencyInfoDetail]
TimbreHourInfoDetail	[TimbreHourInfoDetail]

©2013-2025 Tencent Cloud International Pte. Ltd.

SmallSampleGeneral2DImageAssetInfoDetail	[SmallSampleGeneral2DImageAssetInfoDetail]
SmallSampleGeneral2DInteractConcurrencyInfoDetail	[SmallSampleGeneral2DInteractConcurrencyInfoDetai
SmallSampleGeneral2DBroadcastHourInfoDetail	[SmallSampleGeneral2DBroadcastHourInfoDetail]
SmallSampleGeneral2DBroadcastConcurrencyInfoDetail	[SmallSampleGeneral2DBroadcastConcurrencyInfoDe
RealMan2DInteractConcurrencyInfoDetail	[RealMan2DInteractConcurrencyInfoDetail]
RealMan2DBroadcastHourInfoDetail	[RealMan2DBroadcastHourInfoDetail]
RealMan2DBroadcastConcurrencyInfoDetail	[RealMan2DBroadcastConcurrencyInfoDetail]
UniversalAssetInfoDetail	UniversalAssetInfoDetail

©2013-2025 Tencent Cloud International Pte. Ltd.

UniversalConcurrencyInfoDetail	UniversalConcurrencyInfoDetail
UniversalHourInfoDetail	UniversalHourInfoDetail

InteractConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

BroadcastHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

BroadcastConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

ImageAssetInfoDetail

Parameters	Type	Mandatory	Description
CustomizationNum	int	Yes	Avatar Customization Quantity
RemainNum	int	Yes	Avatar Customization Remaining Quantity

TimbreAssetInfoDetail

Parameters	Type	Mandatory	Description
CustomizationNum	int	Yes	Voice Clone Quantity
RemainNum	int	Yes	Voice Clone Remaining Quantity

©2013-2025 Tencent Cloud International Pte. Ltd.

RealMan3DInteractConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

RealMan3DBroadcastHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

RealMan3DBroadcastConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

TimbreHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Timbre Hourly Package Duration
RemainMilliseconds	int	Yes	Timbre Hourly Package Remaining Duration

SmallSampleGeneral2DImageAssetInfoDetail

Parameters	Type	Mandatory	Description
CustomizationNum	int	Yes	Customization Quantity
RemainNum	int	Yes	Remaining Quantity

SmallSampleGeneral2DInteractConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

SmallSampleGeneral2DBroadcastHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

SmallSampleGeneral2DBroadcastConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

RealMan2DInteractConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

RealMan2DBroadcastHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

RealMan2DBroadcastConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

UniversalAssetInfoDetail

Parameters	Type	Mandatory	Description
CustomizationNum	int	Yes	Customization Quantity

RemainNum	int	Yes	Remaining Quantity

UniversalConcurrencyInfoDetail

Parameters	Type	Mandatory	Description
Concurrency	int	Yes	Concurrency Number

UniversalHourInfoDetail

Parameters	Type	Mandatory	Description
OrderMilliseconds	int	Yes	Broadcast Hourly Package Duration
RemainMilliseconds	int	Yes	Broadcast Hourly Package Remaining Duration

Request Sample

Input parameters:

```
{
    "Header": {},
    "Payload": {
        "ServiceType": "ImageAsset",
        "pageIndex": 1,
        "PageSize": 10
    }
}
```

Output response:

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "ServiceAssetInfos": [
```

```
            {
                "ServiceType": "ImageAsset",
                "ImageAssetInfoDetail": {
                    "CustomizationNum": 10,
                    "RemainNum": 2
                },
                "ExpireDate": "2023-04-01 00:00:00"
            },
            {
                 "ServiceType": "RealMan2DBroadcastHour",
                 "UniversalHourInfoDetail": {
                     "OrderMilliseconds": 36000,
                     "RemainMilliseconds": 36000
                 },
                 "ExpireDate": "2023-04-01 00:00:00"
            }
        ],
        "Total": 1
    }
}
```

Querying Timbre Lists by Pagination

Last updated：2025-01-17 16:34:12

Query customer-owned timbre lists by pagination, including replicated voices and public timbres.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/crmserver/customerassetservice/describetimbre

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameter	Type	Mandatory	Description
PageIndex	int	Yes	Current page
PageSize	int	Yes	Page size (maximum of 100)

Response Parameter

Parameter	Type	Mandatory	Description
TimbreAssets	Array of [TimbreAssetDetail]	Yes	Timbre Asset Information
Total	int	Yes	Totals

TimbreAssetDetail

Parameter	Type	Mandatory	Description
TimbreKey	string	Yes	Digital human timbre code
TimbreName	string	Yes	Digital human timbre name
TimbreSample	string	Yes	Digital human timbre demo sample URL

©2013-2025 Tencent Cloud International Pte. Ltd.

TimbreDesc	string	Yes	Digital human timbre description
SupportVirtualmanTypeCodes	Array of [string]	Yes	Timbre-supported digital human type code, see Appendix 3-Digital Human Type.
SupportDriverTypes	Array of [string]	Yes	Timbre-supported Drive Type 1. Text: Text-driven 2. OriginalVoice: Original voice audio-driven 3. ModulatedVoice: Modulated voice audio-driven
ExpireDate	string	Yes	Validity Period
AudioDuration	int	Yes	Total accumulated duration for calling the audio, in milliseconds.
AudioNum	int	Yes	Total accumulated duration for calling the audio, in milliseconds.
EmotionalCategories	Array of [string]	No	A list of emotional styles supported by the timbre. Data is available only for multi-emotion timbres. See Appendix II for details.
TimbreGender	int	Yes	Timbre Gender; 1: male; 2: female
PurchaseType	string	Yes	Purchase Type, Lease: lease, Custom; customize.
TimbreType	string	Yes	Timbre Type Boutique: boutique SampleBasic: Avatar basic edition SampleFast: Avatar ultra edition
TimbreLanguages	Array of [string]	No	The list of languages supported by timbre, only some timbres support multiple languages, see Appendix 4 - Language List for details.

Request Sample:

```
{
    "Header": {},
```

```
    "Payload": {
        "PageIndex": 1,
        "PageSize": 10
    }
}
```

Response Sample

```
{
    "Header": {
        "Code": 0,
        "Message": "",
        "RequestID": "123"
    },
    "Payload": {
        "TimbreDetails": [
            {
                "TimbreKey": "xxx",
                "TimbreName": "female voice 1",
                "TimbreSample": "url",
                "TimbreDesc": "Natural and graceful",
                "SupportVirtualmanTypeCodes": [
                    "small_sample_2d"
                ],
                "SupportDriverTypes": [
                    "Text",
                    "OriginalVoice"
                ],
                "ExpireDate": "2023-04-01 00:00:00",
                "AudioDuration": 50000,
                "AudioNum": 3,
                "TimbreGender": 2,
                "PurchaseType": "Lease",
                "TimbreType": "Boutique"
            }
        ],
        "Total": 1
    }
}
```

Querying Image Asset Information - Query Anchor

Last updated：2024-07-19 10:19:09

Query image asset information and the assigned anchors.

Calling Protocol

HTTPS + JSON
POST /v2/ivh/crmserver/customerassetservice/getanchor
Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanTypeCode	string	Yes	Digital human type code, see Appendix 3 - Digital Human Type.
AnchorCodes	Array of [string]	No	Digital human anchor code
PageIndex	int	Yes	Current page
PageSize	int	Yes	Page size (maximum of 100)

Response Parameter

Parameters	Type	Mandatory	Description
Virtualmans	Array of [Virtualman]	Yes	Anchor List
Total	int	Yes	Totals

Virtualman

Parameters	Type	Mandatory	Description

AnchorName	string	Yes	Digital human anchor name
AnchorCode	string	Yes	Digital human anchor code
VirtualmanType	string	Yes	Digital human type name
VirtualmanTypeCode	string	Yes	Digital human type code
HeaderImage	string	Yes	Digital human avatar
Tag	string	Yes	Digital human model tags, categorized into three types: 1. Basic 2. Standard 3. Advanced Currently, tags only affect the horizontal and vertical positioning features of the digital human.

Request Sample

```
{
    "Header": {},
    "Payload": {
        "pageIndex": 1,
        "PageSize": 10,
        "VirtualmanTypeCode": "real_man_3d",
        "AnchorCodes": []
    }
}
```

Response Sample

```
{
    "Header": {
        "RequestID": "gz6742f18f16917371370112105",
        "SessionID": "gz6742f18f16917371370112106",
        "DialogID": "",
        "Code": 0,
        "Message": "ok"
    },
```

```
    "Payload": {
        "Virtualmans": [
            {
                "HeaderImage": "https://virtualhuman-cos-test-1251316161.cos.ap-nan
                "VirtualmanTypeCode": "real_man_3d",
                "VirtualmanType": "3D Realistic",
                "Tag": "Basic",
                "AnchorCode": "xiaowei",
                "AnchorName": "Xiaowei"
            }
        ],
        "Total": 1
    }
}
```

Querying Image Asset Information - Querying all Avatars under the Anchor

Last updated：2024-07-19 10:19:57

Query image asset information and retrieve all the avatars assigned under a certain anchor.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/crmserver/customerassetservice/getimagebyanchor

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanTypeCode	string	Yes	Digital human type code, see Appendix 3 - Digital Human Type.
AnchorCode	Array of [string]	No	Digital human anchor code
PageIndex	int	Yes	Current page
PageSize	int	Yes	Page size (maximum of 100)

Response Parameter

Parameters	Type	Mandatory	Description
Virtualmans	Array of [Virtualman]	Yes	The avatars under this anchor
Total	int	Yes	Totals

ImageAssetDetail

Parameters	Type	Mandatory	Description

AnchorName	string	Yes	Digital human anchor name
AnchorCode	string	Yes	Digital human anchor code
VirtualmanType	string	Yes	Digital human type
VirtualmanTypeCode	string	Yes	Digital human type code
VirtualmanKey	string	Yes	Digital human virtualmankey
ClothesName	string	Yes	Digital human clothing name
PoseName	string	Yes	Digital human pose
Resolution	string	Yes	Digital human resolution
HeaderImage	string	Yes	Digital human avatar image URL
PoseImage	string	Yes	Digital human pose image URL
ClothesImage	string	Yes	Digital human clothing image URL
SupportDriverTypes	Array of [string]	Yes	Supported drive types of the digital human Text: Text-driven OriginalVoice: Original voice audio-driven ModulatedVoice: Modulated voice audio-driven
SupportAnchorExtraParams	Array of [SupportAnchorExtraParam]	Yes	Description of configurable parameters for the digital human avatars: Currently only 3D clothing color change parameters are included.
VideoDuration	int64	Yes	Total accumulated duration for calling the digital human, in milliseconds.
VideoNum	int64	Yes	Total number of videos for the digital human
ExpireDate	string	Yes	Validity Period
OriginZoom	string	Yes	Zoom factor, rounded to three

			decimal places
ReferenceVideoSegmentUrl	string	Yes	Select Video Clips

SupportAnchorExtraParam

Parameters	Type	Mandatory	Description
Key	string	Yes	Configuration Item Name
Description	string	Yes	Configuration Item Description
Type	int	Yes	The value type of the configuration item. 1: string; 2: value; 3: boolean
Value	string[]	Yes	Current configuration item; configurable field list

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanTypeCode": "small_sample_2d",
        "AnchorCodes": [
            "wl_pink_skirt_suit_stand"
        ],
        "pageIndex": 1,
        "PageSize": 10
    }
}
```

Response Sample

```
{
    "Header": {
        "RequestID": "bj6ffacfae*****339214923249",
        "SessionID": "bj6ffacfae*****339214923250",
```

```
        "DialogID": "",
        "Code": 0,
        "Message": "ok"
    },
    "Payload": {
        "Total": 2,
        "Virtualmans": [
            {
                "ReferenceVideoSegmentUrl": "https://virtualhuman-cos-test-12513161
                "VirtualmanKey": "xxx",
                "VirtualmanType": "2D Small Sample",
                "ExpireDate": "2024-06-25 15:23:15",
                "SupportDriverTypes": [
                    "Text",
                    "OriginalVoice",
                    "ModulatedVoice"
                ],
                "VirtualmanTypeCode": "small_sample_2d",
                "VideoDuration": 354458,
                "VideoNum": 19,
                "Resolution": "1080x1920",
                "AnchorName": "Misty Pink Skirt Suit",
                "PoseName": "Full body standing pose",
                "OriginZoom": "1.000",
                "HeaderImage": "https://virtualhuman-cos-test-1251316161.cos.ap-nan
                "PoseImage": "https://virtualhuman-cos-test-1251316161.cos.ap-nanji
                "ClothesImage": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ClothesName": "Pink Skirt Suit",
                "AnchorCode": "wl_pink_skirt_suit_stand",
                "SupportAnchorExtraParams": [
                    {
                        "Key": "shirtColor",
                        "Description": "Shirt Color",
                        "Type": 1,
                        "Value": [
                            "colorValue"
                        ]
                    },
                    {
                        "Key": "clotheColor",
                        "Description": "Shirt Color",
                        "Type": 1,
                        "Value": [
                            "colorValue"
                        ]
                    }
                ]
```

```
            },
            {
                "ReferenceVideoSegmentUrl": "https://virtualhuman-cos-test-12513161
                "VirtualmanKey": "xxx",
                "VirtualmanType": "2D Small Sample",
                "ExpireDate": "2024-06-25 15:23:15",
                "SupportDriverTypes": [
                    "Text",
                    "OriginalVoice",
                    "ModulatedVoice"
                ],
                "VirtualmanTypeCode": "small_sample_2d",
                "VideoDuration": 38880,
                "VideoNum": 1,
                "Resolution": "1920x1080",
                "AnchorName": "Misty Pink Skirt Suit",
                "PoseName": "Full body standing pose",
                "OriginZoom": "0.567",
                "HeaderImage": "https://virtualhuman-cos-test-1251316161.cos.ap-nan
                "PoseImage": "https://virtualhuman-cos-test-1251316161.cos.ap-nanji
                "ClothesImage": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ClothesName": "Pink Skirt Suit",
                "AnchorCode": "wl_pink_skirt_suit_stand",
                "SupportAnchorExtraParams": []
            }
        ]
    }
}
```

Querying the List Of Actions Supported by the Avatar

Last updated：2024-07-19 10:20:14

Query the list of supported action information for the avatar based on the virtualmankey.

Note:

This API only supports querying actions for 3D avatars. 2D sample avatars do not support actions, and calling this API will return an empty result.

Calling Protocol

HTTPS + JSON

POST /v2/ivh/crmserver/customerassetservice/getaction

Header Content-Type: application/json;charset=utf-8

Request Parameters

Parameters	Type	Mandatory	Description
VirtualmanKey	string	Yes	Assigned digital human avatar virtualmankey

Response Parameter

Parameters	Type	Mandatory	Description
Actions	Array of [Action]	No	List of supported actions

Action

Parameters	Type	Mandatory	Description
ActionName	string	Yes	Action name of the digital human
ActionCode	string	Yes	Action code of the digital human, corresponding to SSML action tags

ActionSample	string	Yes	Example URL for the digital human actions

Request Sample

```
{
    "Header": {},
    "Payload": {
        "VirtualmanKey": "xxxx"
    }
}
```

Response Sample

```
{
    "Header": {
        "RequestID": "bj1adef008169675180946964 07",
        "SessionID": "bj1adef008169675180946964 08",
        "DialogID": "",
        "Code": 0,
        "Message": "ok"
    },
    "Payload": {
        "Actions": [
            {
                "ActionCode": "2hands_forward1",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Both hands pointing forward 1"
            },
            {
                "ActionCode": "admission",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Entering the stage"
            },
            {
                "ActionCode": "appearance",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Exiting the stage"
            },
            {
                "ActionCode": "bow",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
```

```
            "ActionName": "Bowing"
        },
        {
            "ActionCode": "waving_hand1",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Waving one hand"
        },
        {
            "ActionCode": "welcome",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Welcome"
        },
        {
            "ActionCode": "left_across_waist",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Left hand side slide"
        },
        {
            "ActionCode": "left_side1",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Left hand pointing to the left 1"
        },
        {
            "ActionCode": "left_up2",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Left hand pointing to the left 2"
        },
        {
            "ActionCode": "right_side1",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            ""ActionName": "Right hand pointing to the right 1"
        },
        {
            "ActionCode": "right_slide_down",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Right hand sliding down at the side"
        },
        {
            "ActionCode": "right_up2",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Right hand pointing to the upper right 2"
        },
        {
            "ActionCode": "applaud",
            "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
            "ActionName": "Applauding"
        },
```

```
            {
                "ActionCode": "compliment_state3",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Both hands giving thumbs up"
            },
            {
                "ActionCode": "ok2",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "OK gesture at the side"
            },
            {
                "ActionCode": "show_love3",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Finger heart"
            },
            {
                "ActionCode": "support",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Come on"
            },
            {
                "ActionCode": "turn_around",
                "ActionSample": "https://virtualhuman-cos-test-1251316161.cos.ap-na
                "ActionName": "Turning around"
            }
        ]
    }
}
```

Appendix 1 - Service Asset Type

Last updated：2024-07-19 10:20:27

Parameters	Description
InteractConcurrency	Studio Avatar Interactive Concurrency
RealMan3DInteractConcurrency	3D Realistic Interactive Concurrency
BroadcastHour	Studio Avatar Broadcast Hourly Package
RealMan3DBroadcastHour	3D Realistic Broadcast Hourly Package
BroadcastConcurrency	Studio Avatar Broadcast Concurrency
RealMan3DBroadcastConcurrency	3D Realistic Broadcast Concurrency
ImageAsset	Studio Avatar Image Customization Quota
TimbreAsset	Voice Clone (Basic Edition) Customization Quota
ZeroShotTimbreAsset	Voice Clone (Ultra Edition) Customization Quota
TimbreHour	Timbre Hourly Package
SmallSampleGeneral2DImageAsset	Instant Avatar Image Customization Quota
SmallSampleGeneral2DInteractConcurrency	Instant Avatar Interactive Concurrency
SmallSampleGeneral2DBroadcastHour	Instant Avatar Broadcast Hourly Package
SmallSampleGeneral2DBroadcastConcurrency	Instant Avatar Broadcast Concurrency
RealMan2DInteractConcurrency	2D Premium Interactive Concurrency
RealMan2DBroadcastHour	2D Premium Broadcast Hourly Package
RealMan2DBroadcastConcurrency	2D Premium Broadcast Concurrency
SmallSample4K2DImageAsset	4K Studio Avatar Image Customization Quota
SmallSample4K2DInteractConcurrency	4K Studio Avatar Interactive Concurrency
SmallSample4K2DBroadcastHour	4K Studio Avatar Broadcast Hourly Package
SmallSample4K2DBroadcastConcurrency	4K Studio Avatar Broadcast Concurrency

©2013-2025 Tencent Cloud International Pte. Ltd.

SmallSamplePhoto2DImageAsset	Photo Avatar Image Customization Quota
SmallSamplePhoto2DInteractConcurrency	Photo Avatar Interactive Concurrency
SmallSamplePhoto2DBroadcastHour	Photo Avatar Broadcast Hourly Package
SmallSamplePhoto2DBroadcastConcurrency	Photo Avatar Broadcast Concurrency
SmallSampleImageRenewalAsset	2D Avatar - Exclusive Mouth Shape On-shelf Service Quota
SmallSampleGeneralImageRenewalAsset	2D Avatar - Universal Mouth Shape On-shelf Service Quota
RealMan3DImageRenewalAsset	3D Avatar On-shelf Service Quota
TimbreRenewalAsset	Custom Timbre On-shelf Service Quota

©2013-2025 Tencent Cloud International Pte. Ltd.

Appendix 2 - Emotional Style

Last updated：2024-07-19 10:20:38

Parameters	Description
0	Happy
1	News
2	Customer Service
3	Fear
4	Poetry
5	Story
6	Neutral
7	Sad
8	Broadcast
9	Angry
tc_amaze	Shocked
tc_angry	Angry
tc_aojiao	Tsundere
tc_call	Customer Service
tc_disgusted	Disgust
tc_exciting	Excited
tc_fear	Fear
tc_happy	Happy
tc_jieshuo	Commentary
tc_neutral	Neutral
tc_news	News

©2013-2025 Tencent Cloud International Pte. Ltd.

tc_peaceful	Calm
tc_poetry	Poetry
tc_radio	Broadcast
tc_sad	Sad
tc_sajiao	Childish
tc_story	Story

©2013-2025 Tencent Cloud International Pte. Ltd.

Appendix 3 - Digital Human Type

Last updated：2024-07-19 10:20:51

Parameters	Description
real_man_2d	2D Premium
real_man_3d	3D Realistic
half_realistic_3d	3D Semi-Realistic
cartoon_3d	3D Cartoon
small_sample_2d	Studio Avatar
small_sample_general_2d	Instant Avatar
small_sample_4k_2d	4K Studio Avatar
small_sample_photo_2d	Photo Avatar

©2013-2025 Tencent Cloud International Pte. Ltd.

Appendix 4 - Language List

Last updated：2025-03-12 17:07:11

Parameters	Description
am-ET	Amharic (Ethiopia)
ar-EG	Arabic (Egypt)
ar-SA	Arabic (Saudi Arabia)
bg-BG	Bulgarian (Bulgaria)
bs-BA	Bosnian (Bosnia and Herzegovina)
ca-ES	Catalan
cs-CZ	Czech (Czech Republic)
da-DK	Danish (Denmark)
de-AT	German (Austria)
de-CH	German (Switzerland)
de-DE	German (Germany)
el-GR	Greek (Greece)
en-AU	English (Australia)
en-GB	English (UK)
en-IE	English (Ireland)
en-US	English (US)
es-ES	Spanish (Spain)
es-MX	Spanish (Mexico)
et-EE	Estonian (Estonia)
eu-ES	Basque
fi-FI	Finnish (Finland)

©2013-2025 Tencent Cloud International Pte. Ltd.

fil-PH	Filipino (Philippines)
fr-BE	French (Belgium)
fr-CH	French (Switzerland)
fr-FR	French (France)
gl-ES	Galician
he-IL	Hebrew (Israel)
hr-HR	Croatian (Croatia)
hu-HU	Hungarian (Hungary)
id-ID	Indonesian (Indonesia)
it-IT	Italian (Italy)
ja-JP	Japanese (Japan)
jv-ID	Javanese (Latin, Indonesia)
ko-KR	Korean (South Korea)
mk-MK	Macedonian (north Macedonia)
ms-MY	Malay (Malaysia)
nb-NO	Written Norwegian (Norway)
ne-NP	Nepali (Nepal)
nl-BE	Dutch (Belgium)
nl-NL	Dutch (Netherlands)
pl-PL	Polish (Poland)
ps-AF	Pashto (Afghanistan)
pt-BR	Portuguese (Brazil)
pt-PT	Portuguese (Portugal)
ro-RO	Romanian (Romania)
sk-SK	Slovak (Slovakia)

©2013-2025 Tencent Cloud International Pte. Ltd.

sl-SI	Slovenian (Slovenia)
sv-SE	Swedish (Sweden)
sw-KE	Swahili (Kenya)
th-TH	Thai (Thailand)
tr-TR	Turkish (Türkiye)
uk-UA	Ukrainian (Ukraine)
vi-VN	Vietnamese (Vietnam)
yue-CN	Chinese (Cantonese, simplified)
zh-CN	Chinese (Mandarin, simplified)
zh-HK	Chinese (Cantonese, traditional)
zh-TW	Chinese (Taiwan Mandarin, traditional)
zu-ZA	Zulu (South Africa)

©2013-2025 Tencent Cloud International Pte. Ltd.

API Integration FAQs

Last updated：2024-07-19 10:21:03

Problem Description	Solutions	Notes
Does the personal asset API support querying details of already created video records?	Not supported at the moment. Currently, only the total number of videos and the total hours spent can be queried.	-

©2013-2025 Tencent Cloud International Pte. Ltd.