

ALIBABA CLOUD

Alibaba Cloud

内容安全

API Reference

Document Version: 20201125

 Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions









Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.Development preparations -----	06
2.List of operations by function -----	07
3.Endpoint -----	09
4.Getting started -----	10
4.1. Request structure -----	10
4.2. Common parameters -----	10
4.3. Signature method -----	14
4.4. Common response parameters -----	16
5.RAM user authorization -----	19
6.Image -----	20
6.1. Image moderation -----	20
6.1.1. Moderate images synchronously -----	20
6.1.2. Moderate images asynchronously -----	33
6.1.3. Give feedback on moderation results -----	49
6.2. Image OCR -----	51
6.2.1. Image OCR -----	51
6.2.1.1. Detect text in images synchronously -----	51
6.2.1.2. Detect text in images asynchronously -----	57
7.Video -----	67
7.1. Video moderation -----	67
7.1.1. Moderate videos synchronously -----	67
7.1.2. Moderate videos asynchronously -----	77
7.1.3. Stop moderation tasks -----	97
7.1.4. Give feedback on moderation results -----	98
8.Text -----	102
8.1. Moderate text for violations -----	102

8.2. Give feedback on moderation results	108
--	-----

1. Development preparations

Before you activate Alibaba Cloud Content Moderation, you must familiarize yourself with the service agreement, billing methods, and limits, and complete development preparations.

Get familiar with the service agreement and billing methods


Make sure that you have fully understood the service agreement and billing methods before you use the Content Moderation API.

Get started with the Content Moderation API

Before you use the Content Moderation API, you must register an Alibaba Cloud account, create an AccessKey pair, and activate Alibaba Cloud Content Moderation.

Procedure

1. Go to the [Alibaba Cloud official website](#) and register an account. Skip this step if you have registered an account.
2. Go to the [landing page of Alibaba Cloud Content Moderation](#) and click **Enable Now** to activate the service.
3. Manage your AccessKey ID and AccessKey secret on the [Security Management page](#).

 **Note** An AccessKey pair is the key for using the Content Moderation API and has all permissions on your Alibaba Cloud account. Keep your AccessKey pair confidential and change it periodically.

Get familiar with usage limits

If you encounter the following limits or have the following requests on using the Content Moderation API, you can contact Alibaba Cloud engineers by submitting a ticket. Alibaba Cloud engineers will provide a custom service plan for you based on your requirements. You may have the following requirements:

- The number of objects that you want to moderate at a time exceeds the upper limit.
For more information about concurrency limits, see [Does Content Moderation API impose a concurrency limit?](#)
- You want to moderate a large number of images at a more favorable price.
- You want to use more content moderation services.

2. List of operations by function

The following tables list the API operations available for use in Alibaba Cloud Content Moderation.

Make API requests

This topic applies to users who initiate HTTP or HTTPS POST requests by using request URLs. If you use an SDK to initiate HTTP or HTTPS POST requests, you can skip this topic.

Each request URL consists of different parameters and has a fixed syntax. Generally, a URL contains common parameters, your signature, and operation-specific parameters. Sample request URLs are provided for each operation. These URLs are not encoded to make them easy to read. Before you make a request, you must encode the request URL. A response is returned to you after the authentication is complete based on your signature. Response parameters are returned if the call is successful, whereas an error message is returned if the call fails. You can troubleshoot issues based on the common error codes and operation-specific error codes.


 **Note** We recommend that you use an SDK to call an operation and manage your resources. If you use an SDK, manual signature verification is not required.

Image moderation

Operation	Description
/green/image/scan	Submits image moderation tasks to moderate images for illegal or undesirable content and obtains results in real time.
/green/image/asyncscan	Submits asynchronous image moderation tasks to moderate images for illegal or undesirable content.
/green/image/results	Queries asynchronous image moderation results.
/green/image/feedback	Gives feedback on image moderation results.

Image OCR

Operation	Description
/green/image/scan	Submits optical character recognition (OCR) tasks to detect text in images and obtains results in real time.
/green/image/asyncscan	Submits asynchronous OCR tasks to detect text in images.
/green/image/results	Queries asynchronous OCR results.

Video moderation

Operation	Description
/green/video/syncscan	Submits video moderation tasks to detect illegal or undesirable content in videos and obtains results in real time.
/green/video/asyncscan	Submits asynchronous video moderation tasks to detect illegal or undesirable content in videos.
/green/video/results	Queries asynchronous video moderation results.
/green/video/feedback	Gives feedback on video moderation results.

Text anti-spam

Operation	Description
/green/text/scan	Submits text moderation tasks to detect illegal content in text.
/green/text/feedback	Gives feedback on text moderation results.

3.Endpoint

This topic describes the endpoints of content moderation services.

The endpoint of a content moderation service is in the following format:

```
http(s)://green.{region}.aliyuncs.com/Operation
```

Where,

- `{region}` indicates a supported region. The following table lists the supported regions.

Region	Code	Public network endpoint	Support HTTPS or not	Support internal network access from ECS or not
China (Shanghai)	cn-shanghai	green.cn-shanghai.aliyuncs.com	Supported only by SDK V3.0.0 or later	Not supported
China (Beijing)	cn-beijing	green.cn-beijing.aliyuncs.com	Supported only by SDK V3.0.0 or later	Not supported
China (Shenzhen)	cn-shenzhen	green.cn-shenzhen.aliyuncs.com	Supported only by SDK V3.0.0 or later	Not supported
Singapore	ap-southeast-1	green.ap-southeast-1.aliyuncs.com	Supported only by SDK V3.0.0 or later	Not supported
US (Silicon Valley)	us-west-1	green.us-west-1.aliyuncs.com	Supported only by SDK V3.0.0 or later	Not supported
Indonesia	ap-southeast-5	green.ap-southeast-5	Supported only by SDK V3.0.0 or later	Not supported

- `Operation` indicates the URI of the specific operation. For more information, see the description of each operation.

For example, the following endpoint points to the pornography detection service in the China (Shanghai) region:

```
http(s)://green.cn-shanghai.aliyuncs.com/green/image/scan
```

4. Getting started

4.1. Request structure

This topic describes the structure of the HTTP and HTTPS POST requests sent to call the Content Moderation API.

Methods

The Content Moderation API supports both HTTP and HTTPS POST requests. The JSON-formatted request parameters are sent to the server in the form of an HTTP request body. The common query parameter `clientInfo` is available in the URL query parameters.

The following is an example of an unencoded synchronous request for moderating images:

```
http://green.cn-shanghai.aliyuncs.com/green/image/scan?clientInfo={"userId":"120234234","userNick":"Mike","userType":"others"}
```

Where,

- `http` specifies the protocol for transmitting the request.
- `green.cn-shanghai.aliyuncs.com` specifies the endpoint of the Content Moderation API.
- `green/image/scan` specifies a specific operation, that is, the operation for synchronous image moderation.
- `clientInfo={"ip":"127.0.0.2","userId":"120234234","userNick":"Mike","userType":"others"}` specifies the common query parameters.

Protocols

You can send requests over HTTP or HTTPS. With regard to higher security, we recommend that you send requests over HTTPS, especially when sensitive data is involved.

Parameters

In each request, you must specify [common request parameters](#) and operation-specific service request parameters. Service parameters are transparently transmitted in the request body. For more information about the body structure, see the description of each operation.

Encoding

All requests and responses are encoded in UTF-8.

4.2. Common parameters

Common parameters are parameters that are included in all HTTP requests. Common parameters consist of common request headers and common request parameters.

Common request headers

The following table describes common request headers that must be used when you send HTTP POST requests by using URLs to call the Content Moderation API.

Header	Type	Required	Description
Accept	String	Yes	The data type of the response that can be accepted. Only the JSON type is supported. Set the value to application/json .
Content-Type	String	Yes	The data type of the request body. Only the JSON type is supported. Set the value to application/json .
Content-MD5	String	Yes	The encoded value that is computed based on the request body. To obtain the value, use the content of the request body as a string, apply the MD5 algorithm on the string to obtain an output of binary data, and then encode the output based on the Base64 algorithm.
Date	String	Yes	The time when the request is sent. This header uses the GMT format. Example: <code>Tue, 17 Jan 2017 10:16:36 GMT</code> .
x-acs-version	String	Yes	The version of the Content Moderation API. Set the value to 2018-05-09 .
x-acs-signature-nonce	String	Yes	A unique, random number that is used to prevent replay attacks. We recommend that you use different numbers for different requests.
x-acs-signature-version	String	Yes	The version of the signature encryption algorithm. Set the value to 1.0 .
x-acs-signature-method	String	Yes	The encryption method of the signature string. Set the value to HMAC-SHA1 .
Authorization	String	Yes	The authentication method. The format is <code>"acs" + " " + AccessKeyId + ":" + signature</code> . Parameter description: <ul style="list-style-type: none"> AccessKeyId: the AccessKey ID that you can obtain from the Alibaba Cloud Management Console. For more information about how to obtain the AccessKey ID, see Create an AccessKey pair. signature: the request signature. For more information about the signature encryption algorithm, see Signature method.

Common request parameters

The following table describes common request parameters that can be used when you send HTTP POST requests by using URLs to call the Content Moderation API.

Parameter	Type	Required	Description
clientInfo	String	No	The information about the client that sends the request. The information is serialized from the ClientInfo structure in the JSON format. The information includes the unique machine identifier (UMID) and International Mobile Equipment Identity (IMEI) of the client. For more information, see ClientInfo .

ClientInfo

Parameter	Type	Required	Description
sdkVersion	String	No	The version of the SDK. This parameter is required when an operation is called by using the SDK. For more information, see SDK overview .
cfgVersion	String	No	The version of the configuration. This parameter is required when an operation is called by using the SDK. For more information, see SDK overview .
userType	String	No	The type of the user account. Valid values: <ul style="list-style-type: none"> • taobao: indicates a Taobao account. • others: indicates an account other than a Taobao account.
userId	String	No	The unique ID of the user in the business system. We strongly recommend that you set this parameter for analyzing the violations of the user.
userNick	String	No	The nickname of the user.
avatar	String	No	The URL that can be used to access the profile picture of the user from the Internet. You can pass the value of this parameter to analyze the behaviors of the user.
imei	String	No	The IMEI of the device.
imsi	String	No	The International Mobile Subscriber Identity (IMSI) of the device.
umid	String	No	The UMID of the device.

Parameter	Type	Required	Description
ip	String	No	<p>The public IP address of the user in the business system.</p> <p>If the request is initiated from the device, this parameter can be left empty. If the request is initiated from the server, this parameter is set to the logon IP address of the user or the public IP address of the device.</p> <p>We recommend that you set this parameter for analyzing the violations of the user. If this parameter is not specified in the request, the server attempts to obtain the IP address from the request URL or HTTP request headers.</p>
os	String	No	The operating system of the device. Example: <i>Android 6.0</i> .
channel	String	No	The channel number.
hostAppName	String	No	The name of the host application.
hostPackage	String	No	The name of the host application package.
hostVersion	String	No	The version of the host application.

Sample requests

```
POST /green/image/scan? clientInfo=%7b%22userId%22%3a%22120234234%22%2c%22userNick%22%3a%22Mike%22%2c%22userType%22%3a%22others%22%7d HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-MD5: C+5Y0crpO4sYgC2DNjycug==
Date: Tue, 15 May 2018 10:16:36 GMT
x-acis-version: 2018-05-09
x-acis-signature-nonce: adfkjfaHKslkjf
x-acis-signature-version: 1.0
x-acis-signature-method: HMAC-SHA1
Authorization: acs q23Q9K4PaGi12Wyk:adFff0FFO6sGcDbDNjRtso==
Host: green.cn-shanghai.aliyuncs.com/
Content-Length: 191
{
  "scenes": [
    "porn"
  ],
  "tasks": [
    {
      "dataId": "test2NlnmO$tAON6qYUrtCRgLo-1mwxdi",
      "url": "https://img.alicdn.com/tfs/TB1urBOQFXXXbMXFXXXXXXXXXX-1442-257.png"
    }
  ]
}
```

Parameter description:

- `POST` specifies the request method.
- `/green/image/scan` is the request path.
- `clientInfo=%7b%22userId%22%3a%22120234234%22%2c%22userNick%22%3a%22Mike%22%2c%22userType%22%3a%22others%22%7d` is the encoded common request parameters.
- The part from `Accept:application/json` to `Content-Length:191` are HTTP request headers. In this part, the headers from `Accept: application/json` to `Authorization: acs q23Q9K4PaGi12Wyk:adFff0FFO6sGcDbDNjRtso==` are the common request headers used for signature.
- The content enclosed in braces `{}` is the JSON-formatted request body. The structure of the request body depends on the definition of different API operations. For more information, see [List of operations by function](#).

4.3. Signature method

This topic describes the signature method that you can use to call the Content Moderation API.

Procedure

1. Serialize request headers.

Concatenate all the request headers that start with `x-acs-` to create a string based on the following rules:

- i. Extract all the HTTP headers that start with `x-acs-`.
- ii. Sort the extracted HTTP headers in alphabetical order.
- iii. Write each HTTP header in the format of `"HTTP header name" + ":" + "HTTP header value" + "\n"`.

2. Serialize the URI and query parameters.

Use the `uri + "?clientInfo =" + "JSON string of ClientInfo"` format to concatenate the URI and `clientInfo` parameters.

 **Note** In this step, URL encoding is not required.

3. Create a string-to-sign.

Use the following syntax:


```
"POST\napplication/json\n" +
"Value of the Content-MD5 header" + "\n" +
"application/json" + "\n" +
"Value of the Date header" + "\n" +
"Serialized request headers" +
"Serialized URI and query parameters"
```

The following example shows a complete string-to-sign that can be used to call a synchronous operation for image moderation:


```
POST
application/json
C+5Y0crpO4sYgC2DNjycug==
application/json
Tue, 14 Mar 2017 06:29:50 GMT
x-acs-signature-method:HMAC-SHA1
x-acs-signature-nonce:339497c2-d91f-4c17-a0a3-1192ee9e2202
x-acs-signature-version:1.0
x-acs-version:2018-05-09
/green/image/scan? clientInfo={"ip":"127.xxx.xxx.2","userId":"12023xxxx","userNick":"Mike","userType":
:"others"}
```

4. Generate the signature string.

- i. Use the HMAC-SHA1 algorithm to calculate the HMAC value of the string-to-sign that is created in Step 3. The AccessKey secret is used as the key for the HMAC calculation. The HMAC value is an array of bytes.

 **Note** In this step, do not perform other operations on the HMAC value.

- ii. Encode the HMAC value in Base64 to obtain the signature string.
- iii. Set the value of `signature` to the obtained signature string in the `Authorization` header. The value of the Authorization header follows the syntax: `"acs" + " " + AccessKeyId + ":" + signature`.

 **Note** In the preceding syntax, a space is added between `acs` and `AccessKeyId`.

4.4. Common response parameters

This topic describes the common response parameters and HTTP status codes for calling content moderation operations.

Common response parameters

The following table describes the common parameters in the JSON data returned upon receiving any request.

Parameter	Type	Required	Description
<code>code</code>	Integer	Yes	The HTTP status code returned for the moderation task. Some extended error codes are supported. <ul style="list-style-type: none"> 2xx indicates a success. 4xx indicates an incorrect request. 5xx indicates a back-end error. For more information, see Common HTTP status codes .
<code>msg</code>	String	Yes	The message returned for the moderation task.
<code>requestId</code>	String	Yes	The unique ID of the request, which can be used to locate issues.
<code>data</code>	Integer, String, or JSON object	No	The service-related response data. This parameter may be empty when an error occurs. Generally, the value is a JSON structure or array.

Common HTTP status codes

Identifier	HTTP status code	Description
OK	200	The message returned because the request is successful.
PROCESSING	280	The error message returned because the task is running. We recommend that you wait for a while, for example, 5 seconds, before querying the result.
BAD_REQUEST	400	The error message returned because the request is incorrect. This error is typically caused by incorrect request parameters. Check the request parameters carefully.
NOT_ALLOWED	401	The error message returned because an insecure image or video link is used.
FORBIDDEN	403	The error message returned because your image or video link cannot be accessed. Check whether the link is accessible from the Internet and whether an anti-theft chain policy is configured.
NOT_FOUND	404	The error message returned because the content to be moderated cannot be found. This is usually because your image or video content cannot be downloaded. Make sure that the content is accessible from the Internet.
DOWNLOAD_FAILED	480	The error message returned because the download fails. Make sure that the size and resolution, if any, of the content to be moderated are within the limits of the API.
GENERAL_ERROR	500	The error message returned because a temporary server error occurs. We recommend that you try again. If the error code persists, submit a ticket.
DB_FAILED	580	The error message returned because an error occurs while operating on the database. We recommend that you try again. If the error code persists, submit a ticket.
TIMEOUT	581	The error message returned because the operation times out. We recommend that you try again. If the error code persists, submit a ticket.
CACHE_FAILED	585	The error message returned because a cache error occurs. We recommend that you try again. If the error code persists, submit a ticket.
ALGO_FAILED	586	The error message returned because an algorithm error occurs. Submit a ticket to us.
MQ_FAILED	587	The error message returned because a middleware error occurs. Submit a ticket to us.
EXCEED_QUOTA	588	The error message returned because the number of requests exceeds the quota. The quota defaults to 50 images, 20 videos, 20 audios, or 100 pieces of text per second. If you want to adjust the quota, submit a ticket.

Identifier	HTTP status code	Description
TOO_LARGE	589	The error message returned because the content to be moderated is too large in size. Make sure that the content size is allowed by the API. We recommend that you try again. If the error code persists, submit a ticket.
BAD_FORMAT	590	The error message returned because the content to be moderated is in an incorrect format. Make sure that the format of the content is allowed by the API.
CONNECTION_POOL_FULL	591	The error message returned because the connection pool is full. Submit a ticket to us.
DOWNLOAD_TIMEOUT	592	The error message returned because the download times out. The download duration is limited to 3 seconds. Make sure that the size of the moderated content is allowed by the API.
EXPIRED	594	The error message returned because the task expires. For example, taskId expires.
CATCH_FRAME_FAILED	595	The error message returned because frame capturing fails. Submit a ticket to us.
PERMISSION_DENY	596	The error message returned because the account is unauthorized, overdue, inactivated, or disabled. For more information, see the returned message.

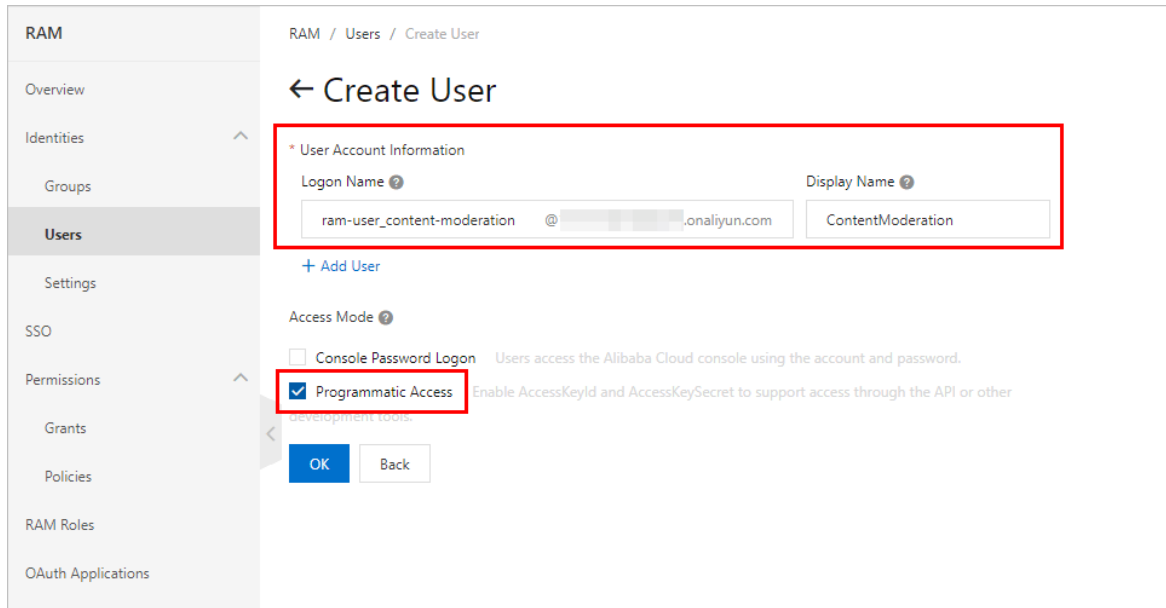
5.RAM user authorization

You can call the Content Moderation API as a RAM user. Before calling the Content Moderation API, make sure that you have created a RAM user and granted relevant permissions to the RAM user.

Procedure

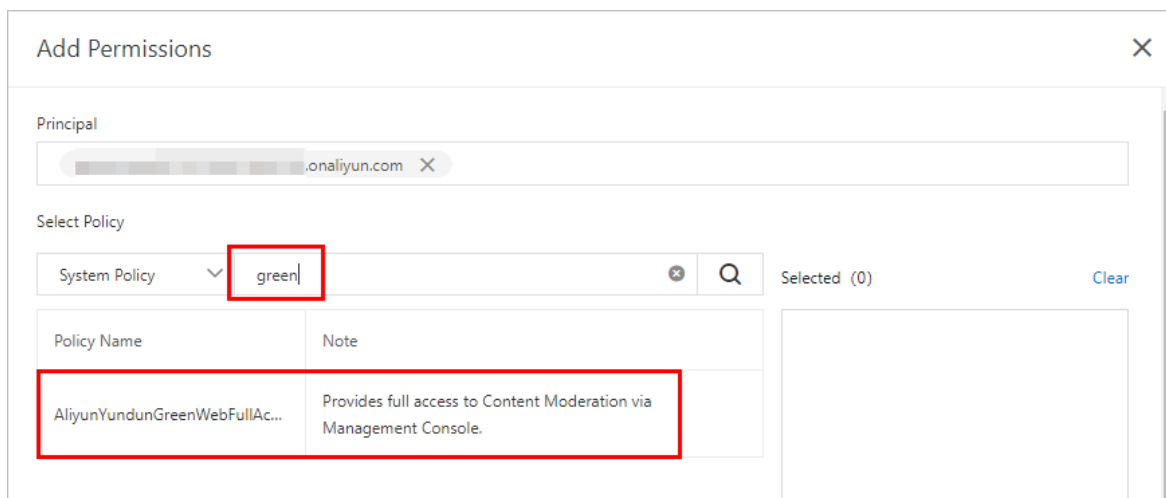
1. Log on to the [RAM console](#), create a RAM user, and then generate the AccessKey ID and AccessKey secret. Properly keep the generated AccessKey ID and AccessKey secret for calling the SDKs later.

For more information about how to create a RAM user, see [Create a RAM user](#).



2. Grant relevant permissions to the RAM user. Only an authorized RAM user can call the Content Moderation API. Grant the following system policy permission to the RAM user you created: `AliyunYundunGreenWebFullAccess`.

For more information about how to grant permissions to a RAM user, see [Grant permissions to a RAM user](#).



6.Image

6.1. Image moderation

6.1.1. Moderate images synchronously

This topic describes the `/green/image/scan` operation that you can call to moderate images for risky and illegal content and obtain moderation results in real time. This operation is applicable to the following scenarios: pornography detection, terrorist content detection, ad violation detection, QR code detection, undesirable scene detection, and logo detection.

Scenarios

Scenario	Description	Category of the moderation result
Pornography detection	Detects pornographic content in images.	normal, sexy, and porn
Terrorist content detection	Detects terrorist content in images.	normal, bloody, explosion, outfit, logo, weapon, politics, violence, crowd, parade, carcrash, flag, location, and others
Ad violation detection	Detects ads or illegal text in images.	normal, ad, politics, porn, abuse, terrorism, contraband, spam, npx, qrcode, and programCode Note By default, you can use only the normal and ad categories. If you want to use other categories, submit a ticket .
QR code detection	Detects QR codes or applet codes in images.	normal, qrcode, and programCode Note By default, you can use only the normal and qrcode categories. If you want to use the programCode category, submit a ticket .
Undesirable scene detection	Detects undesirable scenes, such as black screen, black edge, dark image, picture-in-picture, smoking, and live broadcasting in a running vehicle, in images.	normal, meaningless, PIP, smoking, and driveline
Logo detection	Detects logos, such as TV station logos and trademarks, in images.	normal, TV, and trademark

Description

Operation: `/green/image/scan`

You can call this operation to submit image moderation tasks and obtain moderation results in real time. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

- **Response time:**

The maximum response time that is allowed for a synchronous moderation request is 6s. If the moderation is not complete within 6s, a timeout error is returned. If you do not need to obtain moderation results in real time, you can send asynchronous moderation requests. In most cases, we recommend that you send synchronous moderation requests because synchronous moderation operations are easier to call. We recommend that you set the timeout period to 6s for calling synchronous moderation operations.

- **Return results:**


In general, moderation results are returned within 1s after you send a synchronous moderation request. The time may increase in special scenarios where a large number of requests are to be processed in the system, the size of images is large, or the images contain a large number of words.

- **Limits on images:**


- The images must use HTTP or HTTPS URLs.
- The images must be in the PNG, JPG, JPEG, BMP, GIF, or WEBP format.
- An image can be up to 10 MB in size. The limit for the image size is applicable to both synchronous and asynchronous moderation operations. If you have special requirements, for example, you want to moderate images larger than 10 MB in size, submit a to raise the threshold.
- The duration for downloading an image is limited to 3s. If an image fails to be downloaded within 3s, a timeout error is returned.
- We recommend that you submit images of at least 256 × 256 pixels to ensure the moderation effects.
- The response time of an operation for moderating images varies based on the duration for downloading these images. Make sure that you use a stable and reliable storage service to store the images to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS) or Content Delivery Network (CDN).


Request parameters

Parameter	Type	Required	Description
bizType	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.

Parameter	Type	Required	Description
scenes	String array	Yes	<p>The moderation scenario. Valid values:</p> <ul style="list-style-type: none"> • <i>porn</i>: pornography detection • <i>terrorism</i>: terrorist content detection • <i>ad</i>: ad violation detection • <i>qrcode</i>: QR code detection • <i>live</i>: undesirable scene detection • <i>logo</i>: logo detection <p>You can specify multiple moderation scenarios. For example, you can specify both <i>porn</i> and <i>terrorism</i> in the <code>scenes</code> parameter to detect pornographic and terrorist content.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note If you moderate images in multiple scenarios at a time, you are charged the cumulative fee of all scenarios. The fee of each scenario equals the number of images that are moderated in the scenario multiplied by the unit price of the scenario.</p> </div>
tasks	JSON array	Yes	<p>The list of image moderation tasks. Each element in the JSON array is an image moderation task structure and corresponds to an image. The JSON array can contain a maximum of 10 elements. In other words, you can submit a maximum of 10 images at a time. For more information about the structure of each element, see task.</p>


task

Parameter	Type	Required	Description
clientInfo	JSON structure	No	<p>The information about the client. For more information, see the "Common query parameters" section of the Common parameters topic.</p> <p>The server determines whether to use the global <code>clientInfo</code> parameter or the <code>clientInfo</code> parameter that is described in this table.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note The <code>clientInfo</code> parameter in this table takes priority over the global one.</p> </div>
dataId	String	No	<p>The ID of the image to be moderated. Make sure that each ID is unique in a request.</p>
url	String	Yes	<p>The URL of the image to be moderated.</p>


Parameter	Type	Required	Description
extras	JSON structure	No	The extra parameters that you can specify for calling the operation. The extras parameter is not required for image moderation.
interval	Integer	No	<p>The interval between two frames that are consecutively captured. This parameter is dedicated for GIF or long image moderation.</p> <ul style="list-style-type: none"> A GIF image can be regarded as an array of frames. One frame is captured for moderation from every n frames. The n is specified by the interval parameter. The system captures frames from GIF images only when the interval is specified. Long images can be in portrait or horizontal mode. <ul style="list-style-type: none"> To moderate a long portrait image, in which the height is greater than 400 pixels and the ratio of height to width is greater than 2.5, you can divide the height by the width and round up the result to the nearest integer as the total number of frames. To moderate a long horizontal image, in which the width is greater than 400 pixels and the ratio of width to height is greater than 2.5, you can divide the width by the height and round up the result to the nearest integer as the total number of frames. <p>By default, only the first frame of a GIF image or a long image is moderated. You can use the interval parameter to specify the interval between two frames that the system consecutively captures.</p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfcfcf;"> <p> Note The interval and maxFrames parameters must be used in pairs. For example, the interval parameter is set to 2 and the maxFrames parameter is set to 100 for moderating a GIF image or a long image. In this example, one out of every two frames is moderated and a maximum of 100 frames are moderated. The fee is calculated based on the actual number of moderated frames.</p> </div>

Parameter	Type	Required	Description
maxFrames	Integer	No	<p>The maximum number of frames to be captured. This parameter is dedicated for GIF or long image moderation. Default value: 7.</p> <p>If the value of the <code>interval</code> parameter multiplied by that of the <code>maxFrames</code> parameter is smaller than the total number of frames in a GIF image or a long image, the interval for capturing frames is automatically changed to the integer rounded up from the result of dividing the total number of frames in the image by the value of the <code>maxFrames</code> parameter. This helps improve the overall moderation effects.</p>

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code that is returned for the image moderation task.
msg	String	The message that is returned for the image moderation task.
dataId	String	<p>The ID of the moderation object.</p> <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc;"> <p> Note If you set the <code>dataId</code> parameter in the moderation request, the <code>dataId</code> parameter is returned in the response.</p> </div>
taskId	String	The ID of the image moderation task.
url	String	The URL of the moderated image.
extras	JSON structure	<p>The additional information about the moderated image.</p> <p>If you specify <code>ad</code> in the <code>scenes</code> parameter to detect ad violations, the <code>extras</code> parameter can return the following content:</p> <p><code>hitLibInfo</code>: the information about the custom text library that the detected ad or illegal text in the moderated image hits. This parameter is of the array type. For more information about the structure, see hitLibInfo.</p> <p>Example:</p> <pre style="background-color: #f5f5f5; padding: 5px; border: 1px solid #ccc;">"hitLibInfo":[{"context":"Haokan","libCode":"2144002","libName":"Text pattern library a"}]</pre>
results	Array	The return results of the image moderation task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result .

result

Parameter	Type	Description
scene	String	<p>The moderation scenario of the moderated image, which you specify in the moderation request. Valid values:</p> <ul style="list-style-type: none"> • <i>porn</i>: pornography detection • <i>terrorism</i>: terrorist content detection • <i>ad</i>: ad violation detection • <i>qrcode</i>: QR code detection • <i>live</i>: undesirable scene detection • <i>logo</i>: logo detection •
label	String	<p>The category of the moderation result of the moderated image. Valid values vary based on the specified moderation scenario.</p> <ul style="list-style-type: none"> • If the scenes parameter is set to <i>porn</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain sexy or pornographic content. ◦ <i>sexy</i>: The image contains sexy content. ◦ <i>porn</i>: The image contains pornographic content. • If the scenes parameter is set to <i>terrorism</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain terrorist content. ◦ <i>bloody</i>: The image contains bloody content. ◦ <i>explosion</i>: The image contains explosion and smoke. ◦ <i>outfit</i>: The image contains a special costume. ◦ <i>logo</i>: The image contains a logo. ◦ <i>weapon</i>: The image contains a weapon. ◦ <i>politics</i>: The image contains political content. ◦ <i>violence</i>: The image contains violence. ◦ <i>crowd</i>: The image contains a crowd. ◦ <i>parade</i>: The image contains a parade. ◦ <i>carcrash</i>: The image contains a car accident. ◦ <i>flag</i>: The image contains a flag. ◦ <i>location</i>: The image contains a landmark. ◦ <i>others</i>: The image contains other specified content. • If the scenes parameter is set to <i>ad</i>, the valid values of the label parameter are: <div style="background-color: #e0f2f1; padding: 5px; margin: 5px 0;"> <p> Note By default, only <i>normal</i> and <i>ad</i> can be returned. If you want to use other categories, submit a ticket.</p> </div> <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain ads or illegal text. ◦ <i>ad</i>: The image contains other ads.

Parameter	Type	Description
		<ul style="list-style-type: none"> ◦ <i>politics</i>: The text contains political content. ◦ <i>porn</i>: The text contains pornographic content. ◦ <i>abuse</i>: The text contains abuse. ◦ <i>terrorism</i>: The text contains terrorist content. ◦ <i>contraband</i>: The text contains prohibited content. ◦ <i>spam</i>: The text contains junk content. ◦ <i>npx</i>: The image contains an illegal ad. ◦ <i>qrcode</i>: The image contains a QR code. ◦ <i>programCode</i>: The image contains an applet code. <ul style="list-style-type: none"> • If the scenes parameter is set to <i>qrcode</i>, the valid values of the label parameter are: <div style="background-color: #e0f2f7; padding: 5px; margin: 5px 0;"> <p>? Note By default, programCode cannot be returned. If you want to use this category, submit a ticket.</p> </div> <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain QR codes or applet codes. ◦ <i>qrcode</i>: The image contains a QR code. ◦ <i>programCode</i>: The image contains an applet code. • If the scenes parameter is set to <i>live</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain undesirable scenes. ◦ <i>meaningless</i>: The image is meaningless, such as black or white screen. ◦ <i>PIP</i>: The image contains a small picture. ◦ <i>smoking</i>: The image contains smoking content. ◦ <i>drivelive</i>: The image shows live broadcasting in a running vehicle. • If the scenes parameter is set to <i>logo</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain logos. ◦ <i>TV</i>: The image contains a TV station logo. ◦ <i>trademark</i>: The image contains a trademark.
sublabel	String	<p>The subcategory of the moderation result. If the scenes parameter is set to porn or terrorism, the subcategory of the moderation result can be returned.</p> <p>By default, this parameter is not returned. If you want this parameter to be returned, submit a ticket.</p>
suggestion	String	<p>The machine-assisted moderation result of the moderated image. Valid values:</p> <ul style="list-style-type: none"> • <i>pass</i>: The image does not require further actions. • <i>review</i>: The image contains suspected violations and requires human review. • <i>block</i>: The image contains violations. We recommend that you delete or block the image.

Parameter	Type	Description
rate	Floatin g point	The probability that the moderated image falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the image falls into this category.
frames	Array	The temporary access URL of each frame that is truncated from the moderated image because the image is too long. For more information about the structure, see frame .
hintWordsInfo	Array	<p>The information about the term that the detected ad or illegal text in the moderated image hits. For more information about the structure, see hintWordsInfo.</p> <p>Note This parameter is applicable only to ad violation detection.</p> <p>Example:</p> <pre>"hintWordsInfo":[{"context":"Sensitive words"}]</pre>
qrcodeData	String array	<p>The information about the text that is included in the detected QR code in the moderated image.</p> <p>Note This parameter is applicable only to QR code detection.</p>
programCodeData	Array	<p>The location information about the detected applet code in the moderated image. For more information about the structure, see programCodeData.</p> <p>Note This parameter is applicable only to applet code detection. Make sure that applet code detection is enabled.</p>
logoData	Array	<p>The information about the detected logo in the moderated image. For more information about the structure, see logoData.</p> <p>Note This parameter is applicable only to logo detection.</p>
sfaceData	Array	<p>The information about the detected terrorist content in the moderated image. For more information about the structure, see sfaceData.</p> <p>Note This parameter is applicable only to terrorist content detection.</p>

Parameter	Type	Description
ocrData	String array	The information about the detected complete text in the moderated image. Note By default, this parameter is not returned. If this parameter needs to be returned, submit a ticket .

frame

Parameter	Type	Description
rate	Floatin g point	The confidence level. This parameter is only for reference.
url	String	The temporary access URL of the truncated frame. The URL is valid for 5 minutes.

programCodeData

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the applet code area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the applet code area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the applet code area. Unit: pixels.
h	Floatin g point	The height of the applet code area. Unit: pixels.

logoData

Parameter	Type	Description
type	String	The type of the detected logo. For example, a value of TV indicates the logo of a TV station.
name	String	The name of the detected logo.
x	Floatin g point	The distance between the upper-left corner of the logo area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.

Parameter	Type	Description
y	Floatin g point	The distance between the upper-left corner of the logo area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the logo area. Unit: pixels.
h	Floatin g point	The height of the logo area. Unit: pixels.

sfaceData

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the face area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the face area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the face area. Unit: pixels.
h	Floatin g point	The height of the face area. Unit: pixels.
faces	Array	The information about the recognized face in the moderated image. The array contains the following parameters: <ul style="list-style-type: none"> • name: the name of the recognized face that is detected in the moderated image. The value is a string. • rate: the probability that the detected face in the moderated image hits the recognized face. The value is a floating-point number. • id: the ID of the recognized face. The value is a string.

hitLibInfo

Parameter	Type	Description
context	String	The custom text that the detected ad or illegal text in the moderated image hits.
libCode	String	The code of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.
libName	String	The name of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.

hintWordsInfo

Parameter	Type	Description
context	String	The term that the detected ad or illegal text in the moderated image hits.

Examples

Sample requests

```
{
  "scenes": [
    "porn",
    "terrorism",
    "ad",
    "live",
    "qrcode",
    "logo"
  ],
  "tasks": [
    {
      "dataId": "uuid-xxxx-xxxx-1234",
      "url": "http://xxx.xxx.com/xxx.jpg"
    }
  ]
}
```

Sample success responses

```
{
  "msg": "OK",
  "code": 200,
  "data": [
    {
      "msg": "OK",
      "code": 200,
      "dataId": "uuid-xxxx-xxxx-1234",
      "extras": {
      },
      "results": [
        {
          "rate": 99.63,
          "suggestion": "block",
          "label": "sexy",
          "scene": "porn"
        }
      ]
    }
  ]
}
```

```
},
{
  "label": "politics",
  "rate": 91.54,
  "scene": "terrorism",
  "sfaceData": [
    {
      "faces": [
        {
          "id": "AliFace_0001234",
          "name": "Hit name",
          "rate": 91.54
        }
      ],
      "h": 131,
      "w": 97,
      "x": 49,
      "y": 39
    }
  ],
  "suggestion": "block"
},
{
  "extras": {
    "qrcodes": "http://xxx.xxx.com/0.ZZoliO",
    "npx": "72.01",
    "hitCustomLibCode": "8012345000",
    "hitCustomLibName": "Name of the custom image library",
    "hitLibInfo": [
      {
        "context": "Hit text",
        "libCode": "123456",
        "libName": "Name of the text library"
      }
    ]
  },
  "programCodeData": [
    {
      "w": 402.0,
      "h": 413.0,
```

```
        "x": 11.0,
        "y": 0.0
    }
],
"frames": [
    {
        "rate": 89.85,
        "url": "http://xxx.xxx.com/xxx-0.jpg"
    },
    {
        "rate": 68.06,
        "url": "http://xxx.xxx.com/xxx-1.jpg"
    }
],
"rate": 99.91,
"suggestion": "block",
"label": "ad",
"scene": "ad"
},
{
    "rate": 99.91,
    "suggestion": "block",
    "label": "drug",
    "scene": "live"
},
{
    "qrcodeData": [
        "http://xxx.xxx.com/01ZZOliO"
    ],
    "rate": 99.91,
    "suggestion": "review",
    "label": "qrcode",
    "scene": "qrcode"
},
{
    "logoData": [
        {
            "name": "xxx TV",
            "type": "TV",
            "x": 140,
            "y": 68,
```



```


        "w": 106,
        "h": 106
    }
],
    "rate": 99.9,
    "suggestion": "block",
    "label": "TV",
    "scene": "logo"
}
],
    "taskId": "img4wIjcb7p4wH4lAP3111111-123456",
    "url": "http://xxx.xxx.com/xxx.jpg"
}
],
    "requestId": "69B41AE8-1234-1234-1234-12D395695D2D"
}

```

6.1.2. Moderate images asynchronously

This topic describes the `/green/image/asyncscan` operation that you can call to asynchronously moderate images for risky and illegal content. This operation is applicable to the following scenarios: pornography detection, terrorist content detection, ad violation detection, QR code detection, undesirable scene detection, and logo detection.

Scenarios

Scenario	Description	Category of the moderation result
Pornography detection	Detects pornographic content in images.	normal, sexy, and porn
Terrorist content detection	Detects terrorist content in images.	normal, bloody, explosion, outfit, logo, weapon, politics, violence, crowd, parade, carcrash, flag, location, and others
Ad violation detection	Detects ads or illegal text in images.	normal, ad, politics, porn, abuse, terrorism, contraband, spam, npx, qrcode, and programCode <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <p> Note By default, you can use only the normal and ad categories. If you want to use other categories, submit a ticket.</p> </div>

Scenario	Description	Category of the moderation result
QR code detection	Detects QR codes or applet codes in images.	normal, qrcode, and programCode Note By default, you can use only the normal and qrcode categories. If you want to use the programCode category, submit a ticket .
Undesirable scene detection	Detects undesirable scenes, such as black screen, black edge, dark image, picture-in-picture, smoking, and live broadcasting in a running vehicle, in images.	normal, meaningless, PIP, smoking, and driveline
Logo detection	Detects logos, such as TV station logos and trademarks, in images.	normal, TV, and trademark

Submit asynchronous image moderation tasks

Operation: /green/image/asyncscan

You can call this operation to submit asynchronous image moderation tasks. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

- **Response time:**

The maximum response time that is allowed for a synchronous moderation request is 6s. If the moderation is not complete within 6s, a timeout error is returned. If you do not need to obtain moderation results in real time, you can send asynchronous moderation requests. In most cases, we recommend that you send synchronous moderation requests because synchronous moderation operations are easier to call. We recommend that you set the timeout period to 6s for calling synchronous moderation operations.

- **Return results:**


If you send asynchronous moderation requests, the moderation results are not returned in real time. To obtain moderation results, you can poll the moderation results at regular intervals or enable callback notification. The moderation results are retained for up to 1 hour.

- Enable callback notification to obtain image moderation results: When you submit asynchronous image moderation tasks, you can specify a callback URL for receiving moderation results in the callback parameter of the moderation request. For more information about the callback parameter, see [Request parameters](#).
- Poll image moderation results at regular intervals: You do not need to specify the callback parameter when you submit asynchronous image moderation tasks. After you submit the tasks, you can call the /green/image/results operation to query moderation results. For more information about the /green/image/results operation, see [Query asynchronous image moderation results](#).


- **Limits on images:**

- The images must use HTTP or HTTPS URLs.
- The images must be in the PNG, JPG, JPEG, BMP, GIF, or WEBP format.
- An image can be up to 10 MB in size. The limit for the image size is applicable to both synchronous and asynchronous moderation operations. If you have special requirements, for example, you want to moderate images larger than 10 MB in size, submit a to raise the threshold.
- The duration for downloading an image is limited to 3s. If an image fails to be downloaded within 3s, a timeout error is returned.
- We recommend that you submit images of at least 256 × 256 pixels to ensure the moderation effects.
- The response time of an operation for moderating images varies based on the duration for downloading these images. Make sure that you use a stable and reliable storage service to store the images to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS) or Content Delivery Network (CDN).


Request parameters


Parameter	Type	Required	Description
bizType	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
scenes	String array	Yes	<p>The moderation scenario. Valid values:</p> <ul style="list-style-type: none"> • <i>porn</i>: pornography detection • <i>terrorism</i>: terrorist content detection • <i>ad</i>: ad violation detection • <i>qrcode</i>: QR code detection • <i>live</i>: undesirable scene detection • <i>logo</i>: logo detection <p>You can specify multiple moderation scenarios. For example, you can specify both porn and terrorism in the <code>scenes</code> parameter to detect pornographic and terrorist content.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note If you moderate images in multiple scenarios at a time, you are charged the cumulative fee of all scenarios. The fee of each scenario equals the number of images that are moderated in the scenario multiplied by the unit price of the scenario.</p> </div>

Parameter	Type	Required	Description
callback	String	No	<p>The callback URL for notifying you of asynchronous moderation results. HTTP and HTTPS URLs are supported. If you do not set this parameter, you must poll moderation results at regular intervals.</p> <p>If you set the callback parameter in the moderation request, make sure that the specified HTTP or HTTPS URL meets the following requirements: supports the POST method, uses UTF-8 to encode the transmitted data, and supports the checksum and content parameters. To send moderation results to the specified callback URL, Content Moderation returns the checksum and content parameters in callback notifications based on the following rules and format:</p> <ul style="list-style-type: none"> checksum: the string in the UID + Seed + Content format that is generated by the Secure Hash Algorithm 256 (SHA-256) algorithm. UID indicates the ID of your Alibaba Cloud account. You can query the ID in the Alibaba Cloud Management Console. To prevent data tampering, you can use the SHA-256 algorithm to generate a string when your server receives a callback notification and verify the string against the received checksum parameter. <p>Note UID cannot be the ID of a RAM user.</p> <ul style="list-style-type: none"> content: the JSON-formatted string to be parsed to the callback data in the JSON format. For more information about the format of the content parameter, see the sample success responses of each operation that you can call to query asynchronous moderation results. <p>Note If your server receives a callback notification, the server sends HTTP status code 200 to Content Moderation. If your server fails to receive a callback notification, the server sends other HTTP status codes to Content Moderation. After Content Moderation receives an HTTP status code other than 200, Content Moderation continues to push the callback notification until your server receives it. Content Moderation can push a callback notification repeatedly for up to 16 times. After 16 times, Content Moderation stops pushing the callback notification. In this case, we recommend that you check the status of the callback URL.</p>

Parameter	Type	Required	Description
seed	String	No	<p>A random string that is used to generate a signature for the callback notification request.</p> <p>The string can be up to 64 characters in length and can contain letters, digits, and underscores (_). You can customize this string. It is used to verify the callback notification request when Content Moderation pushes callback notifications to your server.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note This parameter is required if you set the callback parameter.</p> </div>
tasks	JSON array	Yes	<p>The list of image moderation tasks. Each element in the JSON array is an image moderation task structure, namely, each element corresponds to an image. For more information about the structure of each element, see task.</p>

task

Parameter	Type	Required	Description
clientInfo	JSON structure	No	<p>The information about the client. For more information, see the "Common query parameters" section of the Common parameters topic.</p> <p>The server determines whether to use the global clientInfo parameter or the clientInfo parameter that is described in this table.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note The clientInfo parameter in this table takes priority over the global one.</p> </div>
dataId	String	No	The ID of the image to be moderated. Make sure that each ID is unique in a request.
url	String	Yes	The URL of the image to be moderated.
extras	JSON structure	No	The extra parameters that you can specify for calling the operation. The extras parameter is not required for image moderation.

Parameter	Type	Required	Description
interval	Integer	No	<p>The interval between two frames that are consecutively captured. This parameter is dedicated for GIF or long image moderation.</p> <ul style="list-style-type: none"> • A GIF image can be regarded as an array of frames. One frame is captured for moderation from every n frames. The n is specified by the interval parameter. The system captures frames from GIF images only when the interval is specified. • Long images can be in portrait or horizontal mode. <ul style="list-style-type: none"> ◦ To moderate a long portrait image, in which the height is greater than 400 pixels and the ratio of height to width is greater than 2.5, you can divide the height by the width and round up the result to the nearest integer as the total number of frames. ◦ To moderate a long horizontal image, in which the width is greater than 400 pixels and the ratio of width to height is greater than 2.5, you can divide the width by the height and round up the result to the nearest integer as the total number of frames. <p>By default, only the first frame of a GIF image or a long image is moderated. You can use the interval parameter to specify the interval between two frames that the system consecutively captures.</p> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc;"> <p> Note The interval and maxFrames parameters must be used in pairs. For example, the interval parameter is set to 2 and the maxFrames parameter is set to 100 for moderating a GIF image or a long image. In this example, one out of every two frames is moderated and a maximum of 100 frames are moderated. The fee is calculated based on the actual number of moderated frames.</p> </div>

Parameter	Type	Required	Description
maxFrames	Integer	No	<p>The maximum number of frames to be captured. This parameter is dedicated for GIF or long image moderation. Default value: 7.</p> <p>If the value of the <code>interval</code> parameter multiplied by that of the <code>maxFrames</code> parameter is smaller than the total number of frames in a GIF image or a long image, the interval for capturing frames is automatically changed to the integer rounded up from the result of dividing the total number of frames in the image by the value of the <code>maxFrames</code> parameter. This helps improve the overall moderation effects.</p>

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code returned for the image moderation task.
msg	String	The message returned for the image moderation task.
dataId	String	<p>The ID of the moderation object.</p> <p>Note If you set the <code>dataId</code> parameter in the moderation request, the <code>dataId</code> parameter is returned in the response.</p>
taskId	String	The ID of the image moderation task.
url	String	The URL of the moderated image.

Examples

Sample requests

```
{
  "scenes": [
    "porn"
  ],
  "tasks": [
    {
      "dataId": "test4lNSMdggA0c56MMvfYoh4e-1mwpx",
      "url": "https://img.alicdn.com/tfs/TB1urBOQFXXXbMXFXXXXXXXXXX-1442-257.png"
    }
  ]
}
```

Sample success responses

```

{
  "code": 200,
  "msg": "OK",
  "requestId": "95AD868A-F5D2-4AEA-96D4-E0273B8E074C",
  "data": [
    {
      "code": 200,
      "msg": "OK",
      "dataId": "test4lNSMdggA0c56MMvfYoh4e-1mwpx",
      "taskId": "fdd25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695",
      "url": "https://img.alicdn.com/tfs/TB1urBOQFXXXbMXFXXXXXXXXXX-1442-257.png"
    }
  ]
}

```

Query asynchronous image moderation results

Operation: /green/image/results

You can call this operation to query asynchronous image moderation results. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

This operation is free of charge.

- **Response time:**

We recommend that you query moderation results at least 30s after you send an asynchronous moderation request. Content Moderation retains moderation results for up to 4 hours. If you query moderation results after 4 hours, the results are deleted.

Request parameters

Parameter	Type	Required	Description
body	JSON array	Yes	The list of IDs of asynchronous moderation tasks that you want to query. You can specify up to 1,000 task IDs.



Response parameters


Parameter	Type	Description
code	Integer	The HTTP status code returned for the image moderation task.
msg	String	The message returned for the image moderation task.

Parameter	Type	Description
dataId	String	The ID of the moderation object. Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.
taskId	String	The ID of the image moderation task.
url	String	The URL of the moderated image.
extras	JSON structure	The additional information about the moderated image. If you specify <i>ad</i> in the scenes parameter to detect ad violations, the extras parameter can return the following content: hitLibInfo: the information about the custom text library that the detected ad or illegal text in the moderated image hits. This parameter is of the array type. For more information about the structure, see hitLibInfo . Example: <pre>"hitLibInfo": [{"context": "Haokan", "libCode": "2144002", "libName": "Text pattern library a"}]</pre>
results	Array	The return results of the image moderation task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result .

result

Parameter	Type	Description
scene	String	The moderation scenario of the moderated image, which you specify in the moderation request. Valid values: <ul style="list-style-type: none"> <i>porn</i>: pornography detection <i>terrorism</i>: terrorist content detection <i>ad</i>: ad violation detection <i>qrcode</i>: QR code detection <i>live</i>: undesirable scene detection <i>logo</i>: logo detection
		The category of the moderation result of the moderated image. Valid values vary based on the specified moderation scenario. <ul style="list-style-type: none"> If the scenes parameter is set to <i>porn</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The image does not contain sexy or pornographic content. <i>sexy</i>: The image contains sexy content.

Parameter	Type	Description
label	String	<ul style="list-style-type: none"> ◦ <i>porn</i>: The image contains pornographic content. • If the scenes parameter is set to <i>terrorism</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain terrorist content. ◦ <i>bloody</i>: The image contains bloody content. ◦ <i>explosion</i>: The image contains explosion and smoke. ◦ <i>outfit</i>: The image contains a special costume. ◦ <i>logo</i>: The image contains a logo. ◦ <i>weapon</i>: The image contains a weapon. ◦ <i>politics</i>: The image contains political content. ◦ <i>violence</i>: The image contains violence. ◦ <i>crowd</i>: The image contains a crowd. ◦ <i>parade</i>: The image contains a parade. ◦ <i>carcrash</i>: The image contains a car accident. ◦ <i>flag</i>: The image contains a flag. ◦ <i>location</i>: The image contains a landmark. ◦ <i>others</i>: The image contains other specified content. • If the scenes parameter is set to <i>ad</i>, the valid values of the label parameter are: <div style="background-color: #e0f2f1; padding: 5px; margin: 5px 0;"> <p> Note By default, only <i>normal</i> and <i>ad</i> can be returned. If you want to use other categories, submit a ticket.</p> </div> <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain ads or illegal text. ◦ <i>ad</i>: The image contains other ads. ◦ <i>politics</i>: The text contains political content. ◦ <i>porn</i>: The text contains pornographic content. ◦ <i>abuse</i>: The text contains abuse. ◦ <i>terrorism</i>: The text contains terrorist content. ◦ <i>contraband</i>: The text contains prohibited content. ◦ <i>spam</i>: The text contains junk content. ◦ <i>npx</i>: The image contains an illegal ad. ◦ <i>qrcode</i>: The image contains a QR code. ◦ <i>programCode</i>: The image contains an applet code. • If the scenes parameter is set to <i>qrcode</i>, the valid values of the label parameter are: <div style="background-color: #e0f2f1; padding: 5px; margin: 5px 0;"> <p> Note By default, programCode cannot be returned. If you want to use this category, submit a ticket.</p> </div> <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain QR codes or applet codes. ◦ <i>qrcode</i>: The image contains a QR code. ◦ <i>programCode</i>: The image contains an applet code. • If the scenes parameter is set to <i>live</i>, the valid values of the label

Parameter	Type	Description
		<p>parameter are:</p> <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain undesirable scenes. ◦ <i>meaningless</i>: The image is meaningless, such as black or white screen. ◦ <i>PIP</i>: The image contains a small picture. ◦ <i>smoking</i>: The image contains smoking content. ◦ <i>drivelive</i>: The image shows live broadcasting in a running vehicle. • If the scenes parameter is set to <i>logo</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ <i>normal</i>: The image does not contain logos. ◦ <i>TV</i>: The image contains a TV station logo. ◦ <i>trademark</i>: The image contains a trademark.
sublabel	String	<p>The subcategory of the moderation result. If the scenes parameter is set to porn or terrorism, the subcategory of the moderation result can be returned.</p> <p>By default, this parameter is not returned. If you want this parameter to be returned, submit a ticket.</p>
suggestion	String	<p>The machine-assisted moderation result of the moderated image. Valid values:</p> <ul style="list-style-type: none"> • <i>pass</i>: The image does not require further actions. • <i>review</i>: The image contains suspected violations and requires human review. • <i>block</i>: The image contains violations. We recommend that you delete or block the image.
rate	Floatin g point	<p>The probability that the moderated image falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the image falls into this category.</p>
frames	Array	<p>The temporary access URL of each frame that is truncated from the moderated image because the image is too long. For more information about the structure, see frame.</p>
hintWordsInfo	Array	<p>The information about the term that the detected ad or illegal text in the moderated image hits. For more information about the structure, see hintWordsInfo.</p> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc;"> <p> Note This parameter is applicable only to ad violation detection.</p> </div> <p>Example:</p> <pre style="background-color: #f5f5f5; padding: 5px; border: 1px solid #ccc;">"hintWordsInfo":[{"context":"Sensitive words"}]</pre>

Parameter	Type	Description
qrcodeData	String array	The information about the text that is included in the detected QR code in the moderated image. Note This parameter is applicable only to QR code detection.
programCodeData	Array	The location information about the detected applet code in the moderated image. For more information about the structure, see programCodeData . Note This parameter is applicable only to applet code detection. Make sure that applet code detection is enabled.
logoData	Array	The information about the detected logo in the moderated image. For more information about the structure, see logoData . Note This parameter is applicable only to logo detection.
sfaceData	Array	The information about the detected terrorist content in the moderated image. For more information about the structure, see sfaceData . Note This parameter is applicable only to terrorist content detection.
ocrData	String array	The information about the detected complete text in the moderated image. Note By default, this parameter is not returned. If this parameter needs to be returned, submit a ticket .

frame

Parameter	Type	Description
rate	Floatin g point	The confidence level. This parameter is only for reference.
url	String	The temporary access URL of the truncated frame. The URL is valid for 5 minutes.

programCodeData

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the applet code area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the applet code area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the applet code area. Unit: pixels.
h	Floatin g point	The height of the applet code area. Unit: pixels.

logoData

Parameter	Type	Description
type	String	The type of the detected logo. For example, a value of TV indicates the logo of a TV station.
name	String	The name of the detected logo.
x	Floatin g point	The distance between the upper-left corner of the logo area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the logo area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the logo area. Unit: pixels.
h	Floatin g point	The height of the logo area. Unit: pixels.

sfaceData

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the face area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the face area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the face area. Unit: pixels.

Parameter	Type	Description
h	Floatin g point	The height of the face area. Unit: pixels.
faces	Array	The information about the recognized face in the moderated image. The array contains the following parameters: <ul style="list-style-type: none"> • name: the name of the recognized face that is detected in the moderated image. The value is a string. • rate: the probability that the detected face in the moderated image hits the recognized face. The value is a floating-point number. • id: the ID of the recognized face. The value is a string.

hitLibInfo

Parameter	Type	Description
context	String	The custom text that the detected ad or illegal text in the moderated image hits.
libCode	String	The code of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.
libName	String	The name of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.

hintWordsInfo

Parameter	Type	Description
context	String	The term that the detected ad or illegal text in the moderated image hits.

Examples

Sample requests

```
[
  "fdd25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695"
]
```

Sample success responses

```
{
  "msg": "OK",
  "code": 200,
  "data": [
    {
      "msg": "OK",
      "code": 200
    }
  ]
}
```

```
"code": 200,
"dataId": "test4lNSMdggA0c56MMvfYoh4e-1mwxpx",
"extras": {
},
"results": [
  {
    "rate": 99.63,
    "suggestion": "block",
    "label": "sexy",
    "scene": "porn"
  },
  {
    "label": "politics",
    "rate": 91.54,
    "scene": "terrorism",
    "sfaceData": [
      {
        "faces": [
          {
            "id": "AliFace_0001234",
            "name": "Hit name",
            "rate": 91.54
          }
        ]
      },
      {
        "h": 131,
        "w": 97,
        "x": 49,
        "y": 39
      }
    ],
    "suggestion": "block"
  },
  {
    "extras": {
      "qrCodes": "http://xxx.xxx.com/0.ZZ0liO",
      "npX": "72.01",
      "hitCustomLibCode": "8012345000",
      "hitCustomLibName": "Name of the custom image library",
      "hitLibInfo": [
        {
          "context": "Hit text",

```

```
        "libCode": "123456",
        "libName": "Name of the text library"
    }
]
},
"programCodeData": [
    {
        "w": 402.0,
        "h": 413.0,
        "x": 11.0,
        "y": 0.0
    }
],
"frames": [
    {
        "rate": 89.85,
        "url": "http://xxx.xxx.com/xxx-0.jpg"
    },
    {
        "rate": 68.06,
        "url": "http://xxx.xxx.com/xxx-1.jpg"
    }
],
"rate": 99.91,
"suggestion": "block",
"label": "ad",
"scene": "ad"
},
{
    "rate": 99.91,
    "suggestion": "block",
    "label": "drug",
    "scene": "live"
},
{
    "qrcodeData": [
        "http://xxx.xxx.com/01ZZOliO"
    ],
    "rate": 99.91,
    "suggestion": "review",
    "label": "arcode".
```



```
        "scene": "qrcode"
      },
      {
        "logoData": [
          {
            "name": "xxx TV",
            "type": "TV",
            "x": 140,
            "y": 68,
            "w": 106,
            "h": 106
          }
        ],
        "rate": 99.9,
        "suggestion": "block",
        "label": "TV",
        "scene": "logo"
      }
    ],
    "taskId": "fdd25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695",
    "url": "https://img.alicdn.com/tfs/TB1urBOQFXXXXbMXFXXXXXXXXXX-1442-257.png"
  }
],
"requestId": "69B41AE8-1234-1234-1234-12D395695D2D"
}
```

6.1.3. Give feedback on moderation results

If you find that image moderation results returned by Alibaba Cloud Content Moderation do not meet your expectations, you can call an operation to provide feedback. Your suggestion will take effect the next time you moderate the same image.

Description

Operation: `/green/image/feedback`



You can call this operation to give feedback on the result of an image moderation task and provide your expected result.

 **Note** You are not charged for calling this operation.

Request parameters

For more information about the common request parameters that must be included in all Content Moderation API requests, see [Common request parameters](#).

The request body is a JSON structure. The following table describes the parameters that the JSON structure contains.

Parameter	Type	Required	Description
taskId	String	No	The ID returned by the Content Moderation server, which uniquely identifies the moderation task.
suggestion	String	No	<p>The moderation result you expect to return. If you specify this parameter, you must specify the scenes parameter. Valid values:</p> <ul style="list-style-type: none"> <i>pass</i>: The image is normal. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Note The image whose moderation result is <i>pass</i> can be automatically added to the whitelist of the feedback-based image library in the Alibaba Cloud Content Moderation console.</p> </div> <ul style="list-style-type: none"> <i>block</i>: The image contains violations.
url	String	No	The URL for adding the moderated image to the feedback-based image library in the Alibaba Cloud Content Moderation console. To add the image to the feedback-based image library, you must specify this parameter.
scenes	String array	No	<p>The scenario of image moderation. Valid values:</p> <ul style="list-style-type: none"> <i>porn</i>: pornography detection <i>terrorism</i>: terrorist content detection <i>ad</i>: ad violation detection <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Note You can specify multiple moderation scenarios. For example, you can specify both <i>porn</i> and <i>terrorism</i> in the <code>scenes</code> parameter to indicate pornography detection and terrorist content detection scenarios.</p> </div>
note	String	No	The description of the moderated image.
label	String	No	The expected category of moderation results for the moderated image in the specified moderation scenario. For more information about valid values, see Parameters scenes and label .

Response parameters

The data parameter in the response body is empty. For more information about common response parameters that this operation returns, see [Common response parameters](#).

Examples

Sample requests

```
{
  "suggestion": "block",
  "scenes": [
    "ad",
    "terrorism"
  ],
  "url": "http://xxx.xxx.xxx.com"
}
```

Sample success responses

```
{
  "code": 200,
  "msg": "OK",
  "requestId": "EE5A1189-4D7B-4C24-AD78-4C1FAA3E7A0C"
}
```

6.2. Image OCR

6.2.1. Image OCR

6.2.1.1. Detect text in images synchronously

This topic describes the `/green/image/scan` operation that you can call to submit optical character recognition (OCR) tasks and obtain OCR results in real time. You can submit the OCR tasks to detect and obtain text in images.

Description

Operation: `/green/image/scan`

You can call this operation to submit synchronous OCR tasks. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

- **Response time:**

The maximum response time that is allowed for a synchronous moderation request is 6s. If the moderation is not complete within 6s, a timeout error is returned. If you do not need to obtain moderation results in real time, you can send asynchronous moderation requests. In most cases, we recommend that you send synchronous moderation requests because synchronous moderation operations are easier to call. We recommend that you set the timeout period to 6s for calling synchronous moderation operations.

- **Return results:**

In general, moderation results are returned within 1s after you send a synchronous moderation request. The time may increase in special scenarios where a large number of requests are to be processed in the system, the size of images is large, or the images contain a large number of words. The speed of OCR is inversely proportional to the number of words in images. If the images to be moderated contain a large number of words, we recommend that you send asynchronous moderation requests.


- **Limits on images:**

- The images must use HTTP or HTTPS URLs.
- The images must be in the PNG, JPG, JPEG, BMP, GIF, or WEBP format.
- An image can be up to 10 MB in size. The limit for the image size is applicable to both synchronous and asynchronous moderation operations. If you have special requirements, for example, you want to moderate images larger than 10 MB in size, submit a to raise the threshold.
- The duration for downloading an image is limited to 3s. If an image fails to be downloaded within 3s, a timeout error is returned.
- We recommend that you submit images of at least 256 × 256 pixels to ensure the moderation effects.
- The response time of an operation for moderating images varies based on the duration for downloading these images. Make sure that you use a stable and reliable storage service to store the images to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS) or Content Delivery Network (CDN).

Request parameters


Parameter	Type	Required	Description
bizType	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
scenes	String array	Yes	The moderation scenario. Set the value to <i>ocr</i> .
tasks	JSON array	Yes	The list of OCR tasks. Each element in the JSON array is an OCR task structure, namely, each element corresponds to an image. The JSON array can contain a maximum of 10 elements, namely, you can submit a maximum of 10 images at a time. For more information about the structure of each element, see task .

task

Parameter	Type	Required	Description
dataId	String	No	The ID of the image to be moderated. Make sure that each ID is unique in a request.
url	String	Yes	The URL of the image to be moderated.
interval	Integer	No	<p>The interval between two frames that are consecutively captured. This parameter is dedicated for GIF or long image moderation.</p> <ul style="list-style-type: none"> • A GIF image can be regarded as an array of frames. One frame is captured for moderation from every n frames. The n is specified by the interval parameter. The system captures frames from GIF images only when the interval is specified. • Long images can be in portrait or horizontal mode. <ul style="list-style-type: none"> ◦ To moderate a long portrait image, in which the height is greater than 400 pixels and the ratio of height to width is greater than 2.5, you can divide the height by the width and round up the result to the nearest integer as the total number of frames. ◦ To moderate a long horizontal image, in which the width is greater than 400 pixels and the ratio of width to height is greater than 2.5, you can divide the width by the height and round up the result to the nearest integer as the total number of frames. <p>By default, only the first frame of a GIF image or a long image is moderated. You can use the interval parameter to specify the interval between two frames that the system consecutively captures.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note The interval and maxFrames parameters must be used in pairs. For example, the interval parameter is set to 2 and the maxFrames parameter is set to 100 for moderating a GIF image or a long image. In this example, one out of every two frames is moderated and a maximum of 100 frames are moderated. The fee is calculated based on the actual number of moderated frames.</p> </div>

Parameter	Type	Required	Description
maxFrames	Integer	No	<p>The maximum number of frames to be captured. This parameter is dedicated for GIF or long image moderation. Default value: 7.</p> <p>If the value of the <code>interval</code> parameter multiplied by that of the <code>maxFrames</code> parameter is smaller than the total number of frames in a GIF image or a long image, the interval for capturing frames is automatically changed to the integer rounded up from the result of dividing the total number of frames in the image by the value of the <code>maxFrames</code> parameter. This helps improve the overall moderation effects.</p>

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code returned for the OCR task.
msg	String	The message returned for the OCR task.
dataId	String	<p>The ID of the moderation object.</p> <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc;"> <p> Note If you set the <code>dataId</code> parameter in the moderation request, the <code>dataId</code> parameter is returned in the response.</p> </div>
taskId	String	The ID of the OCR task.
url	String	The URL of the moderated image.
results	Array	The return results of the OCR task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result .

result

Parameter	Type	Description
scene	String	The moderation scenario. The value is fixed to <code>ocr</code> .
label	String	<p>The category of the OCR results. Valid values:</p> <ul style="list-style-type: none"> <code>normal</code>: The image does not contain text. <code>ocr</code>: The image contains text.

Parameter	Type	Description
suggestion	String	The machine-assisted moderation result of the moderated image. Valid values: <ul style="list-style-type: none"> <i>pass</i>: The image does not require further actions. <i>review</i>: The image requires human review.
rate	Floatin g point	The probability that the moderated image falls into the detected category. You can ignore this parameter in the OCR scenario.
ocrLocations	Array	The information about the single text entry in the moderated static image, which includes the text, text size, and text location. For more information about the structure, see ocrLocation .
ocrData	Array	The combination of all text in the moderated static image. In general, the text combination is stored as the first element of the array.
frames	Array	The frames that are captured from the moderated GIF image and the text that is detected in each frame.

ocrLocation

Parameter	Type	Description
text	String	The single text entry that is detected in the moderated image.
x	Floatin g point	The distance between the upper-left corner of the text area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the text area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the text area. Unit: pixels.
h	Floatin g point	The height of the text area. Unit: pixels.

Examples

Sample requests

```
{
  "scenes": [
    "ocr"
  ],
  "tasks": [
    {
      "dataId": "test_data_xxxx",
      "url": "https://test_image_xxxx.png"
    }
  ]
}
```

Sample success responses

```
{
  "code": 200,
  "data": [
    {
      "code": 200,
      "dataId": "test_data_xxxx",
      "extras": {
      },
      "msg": "OK",
      "results": [
        {
          "label": "ocr",
          "ocrData": [
            "hello, this is a test text."
          ],
          "ocrLocations": [
            {
              "h": 26,
              "text": "hello",
              "w": 83,
              "x": 41,
              "y": 84
            },
            {
              "h": 25,
              "text": " this is a test text.",
              "w": 95,
```



```
        "x": 78,
        "y": 114
      }
    ],
    "rate": 99.91,
    "scene": "ocr",
    "suggestion": "review"
  }
],
"taskId": "img5A@k7a@B4q@6K@d9nfKgOs-1sWeLu",
"url": "https://test_image_xxxx.png"
}
],
"msg": "OK",
"requestId": "C4AB08A9-AD75-4410-859B-0B9EF6DFC3C4"
}
```

6.2.1.2. Detect text in images asynchronously

This topic describes the `/green/image/asyncscan` operation that you can call to submit asynchronous optical character recognition (OCR) tasks. You can submit the OCR tasks to detect and obtain text in images.

Submit asynchronous OCR tasks

Operation: `/green/image/asyncscan`

You can call this operation to submit asynchronous OCR tasks. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

- **Response time:**

The maximum response time that is allowed for a synchronous moderation request is 6s. If the moderation is not complete within 6s, a timeout error is returned. If you do not need to obtain moderation results in real time, you can send asynchronous moderation requests. In most cases, we recommend that you send synchronous moderation requests because synchronous moderation operations are easier to call. We recommend that you set the timeout period to 6s for calling synchronous moderation operations.



- **Return results:**


If you send asynchronous moderation requests, the moderation results are not returned in real time. To obtain moderation results, you can poll the moderation results at regular intervals or enable callback notification. The moderation results are retained for up to 1 hour.

- Enable callback notification to obtain OCR results: When you submit asynchronous OCR tasks, you can specify a callback URL for receiving OCR results in the callback parameter of the OCR request. For more information about the callback parameter, see [Request parameters](#).
- Poll OCR results at regular intervals: You do not need to specify the callback parameter when you submit asynchronous OCR tasks. After you submit the tasks, you can call the /green/image/results operation to query OCR results. For more information about the /green/image/results operation, see [Query asynchronous OCR results](#).
- **Limits on images:**
 - The images must use HTTP or HTTPS URLs.
 - The images must be in the PNG, JPG, JPEG, BMP, GIF, or WEBP format.
 - An image can be up to 10 MB in size. The limit for the image size is applicable to both synchronous and asynchronous moderation operations. If you have special requirements, for example, you want to moderate images larger than 10 MB in size, submit a to raise the threshold.
 - The duration for downloading an image is limited to 3s. If an image fails to be downloaded within 3s, a timeout error is returned.
 - We recommend that you submit images of at least 256 × 256 pixels to ensure the moderation effects.
 - The response time of an operation for moderating images varies based on the duration for downloading these images. Make sure that you use a stable and reliable storage service to store the images to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS) or Content Delivery Network (CDN).

Request parameters


Parameter	Type	Required	Description
bizType	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
scenes	String array	Yes	The moderation scenario. Set the value to <i>ocr</i> .

Parameter	Type	Required	Description
callback	String	No	<p>The callback URL for notifying you of asynchronous moderation results. HTTP and HTTPS URLs are supported. If you do not set this parameter, you must poll moderation results at regular intervals.</p> <p>If you set the callback parameter in the moderation request, make sure that the specified HTTP or HTTPS URL meets the following requirements: supports the POST method, uses UTF-8 to encode the transmitted data, and supports the checksum and content parameters. To send moderation results to the specified callback URL, Content Moderation returns the checksum and content parameters in callback notifications based on the following rules and format:</p> <ul style="list-style-type: none"> checksum: the string in the <code>UID + Seed + Content</code> format that is generated by the Secure Hash Algorithm 256 (SHA-256) algorithm. UID indicates the ID of your Alibaba Cloud account. You can query the ID in the Alibaba Cloud Management Console. To prevent data tampering, you can use the SHA-256 algorithm to generate a string when your server receives a callback notification and verify the string against the received checksum parameter. <div style="background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Note UID cannot be the ID of a RAM user.</p> </div> <ul style="list-style-type: none"> content: the JSON-formatted string to be parsed to the callback data in the JSON format. For more information about the format of the content parameter, see the sample success responses of each operation that you can call to query asynchronous moderation results. <div style="background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Note If your server receives a callback notification, the server sends HTTP status code 200 to Content Moderation. If your server fails to receive a callback notification, the server sends other HTTP status codes to Content Moderation. After Content Moderation receives an HTTP status code other than 200, Content Moderation continues to push the callback notification until your server receives it. Content Moderation can push a callback notification repeatedly for up to 16 times. After 16 times, Content Moderation stops pushing the callback notification. In this case, we recommend that you check the status of the callback URL.</p> </div>

Parameter	Type	Required	Description
seed	String	No	<p>A random string that is used to generate a signature for the callback notification request.</p> <p>The string can be up to 64 characters in length and can contain letters, digits, and underscores (_). You can customize this string. It is used to verify the callback notification request when Content Moderation pushes callback notifications to your server.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note This parameter is required if you set the callback parameter.</p> </div>
tasks	JSON array	Yes	<p>The list of OCR tasks. Each element in the JSON array is an OCR task structure and corresponds to an image. The JSON array can contain a maximum of 100 elements. In other words, you can submit a maximum of 100 images at a time. For more information about the structure of each element, see task.</p>

task

Parameter	Type	Required	Description
dataId	String	No	The ID of the image to be moderated. Make sure that each ID is unique in a request.
url	String	Yes	The URL of the image to be moderated.

Parameter	Type	Required	Description
interval	Integer	No	<p>The interval between two frames that are consecutively captured. This parameter is dedicated for GIF or long image moderation.</p> <ul style="list-style-type: none"> • A GIF image can be regarded as an array of frames. One frame is captured for moderation from every n frames. The n is specified by the interval parameter. The system captures frames from GIF images only when the interval is specified. • Long images can be in portrait or horizontal mode. <ul style="list-style-type: none"> ◦ To moderate a long portrait image, in which the height is greater than 400 pixels and the ratio of height to width is greater than 2.5, you can divide the height by the width and round up the result to the nearest integer as the total number of frames. ◦ To moderate a long horizontal image, in which the width is greater than 400 pixels and the ratio of width to height is greater than 2.5, you can divide the width by the height and round up the result to the nearest integer as the total number of frames. <p>By default, only the first frame of a GIF image or a long image is moderated. You can use the interval parameter to specify the interval between two frames that the system consecutively captures.</p> <div style="background-color: #e0f2f1; padding: 10px; border: 1px solid #ccc;"> <p> Note The interval and maxFrames parameters must be used in pairs. For example, the interval parameter is set to 2 and the maxFrames parameter is set to 100 for moderating a GIF image or a long image. In this example, one out of every two frames is moderated and a maximum of 100 frames are moderated. The fee is calculated based on the actual number of moderated frames.</p> </div>

Parameter	Type	Required	Description
maxFrames	Integer	No	<p>The maximum number of frames to be captured. This parameter is dedicated for GIF or long image moderation. Default value: 7.</p> <p>If the value of the <code>interval</code> parameter multiplied by that of the <code>maxFrames</code> parameter is smaller than the total number of frames in a GIF image or a long image, the interval for capturing frames is automatically changed to the integer rounded up from the result of dividing the total number of frames in the image by the value of the <code>maxFrames</code> parameter. This helps improve the overall moderation effects.</p>

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code that is returned for the OCR task.
msg	String	The message that is returned for the OCR task.
dataId	String	<p>The ID of the moderation object.</p> <p>Note If you set the <code>dataId</code> parameter in the moderation request, the <code>dataId</code> parameter is returned in the response.</p>
taskId	String	The ID of the OCR task.
url	String	The URL of the moderated image.
extras	JSON structure	<p>The extra parameters that you set in the <code>extras</code> parameter of the moderation request.</p> <p>Note This parameter may be subject to changes. Use the latest value of this parameter.</p>

Examples

Sample requests

```
{
  "scenes": [
    "ocr"
  ],
  "tasks": [
    {
      "dataId": "test_data_xxxx",
      "url": "https://test_image_xxxx.png"
    }
  ]
}
```

Sample success responses

```
{
  "code": 200,
  "msg": "OK",
  "requestId": "92AD868A-F5D2-4AEA-96D4-E1273B8E074C",
  "data": [
    {
      "code": 200,
      "msg": "OK",
      "dataId": "test_data_xxxx",
      "taskId": "aaa25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695",
      "url": "https://test_image_xxxx.png"
    }
  ]
}
```

Query asynchronous OCR results

Operation: /green/image/results

You can call this operation to query asynchronous OCR results. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

This operation is free of charge.

- **Response time:**

We recommend that you query moderation results at least 30s after you send an asynchronous moderation request. Content Moderation retains moderation results for up to 4 hours. If you query moderation results after 4 hours, the results are deleted.

Request parameters

Parameter	Type	Required	Description
body	JSON array	Yes	The list of IDs of asynchronous moderation tasks that you want to query. You can specify up to 1,000 task IDs.

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code that is returned for the OCR task.
msg	String	The message that is returned for the OCR task.
dataId	String	The ID of the moderation object. <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;">? Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.</div>
taskId	String	The ID of the OCR task.
url	String	The URL of the moderated image.
results	Array	The return results of the OCR task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result .

result

Parameter	Type	Description
scene	String	The moderation scenario. The value is fixed to <i>ocr</i> .
label	String	The category of the OCR results. Valid values: <ul style="list-style-type: none"> <i>normal</i>: The image does not contain text. <i>ocr</i>: The image contains text.
suggestion	String	The machine-assisted moderation result of the moderated image. Valid values: <ul style="list-style-type: none"> <i>pass</i>: The image does not require further actions. <i>review</i>: The image requires human review.
rate	Floatin g point	The probability that the moderated image falls into the detected category. You can ignore this parameter in the OCR scenario.
ocrLocations	Array	The information about the single text entry in the moderated static image, which includes the text, text size, and text location. For more information about the structure, see ocrLocation .

Parameter	Type	Description
ocrData	Array	The combination of all text in the moderated static image. In general, the text combination is stored as the first element of the array.
frames	Array	The frames that are captured from the moderated GIF image and the text that is detected in each frame.

ocrLocation

Parameter	Type	Description
text	String	The single text entry that is detected in the moderated image.
x	Floatin g point	The distance between the upper-left corner of the text area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the text area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the text area. Unit: pixels.
h	Floatin g point	The height of the text area. Unit: pixels.

Examples

Sample requests

```
[  
  "aaa25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695"  
]
```

Sample success responses

```
{
  "code": 200,
  "data": [
    {
      "code": 200,
      "dataId": "test_data_xxxx",
      "extras": {
      },
      "msg": "OK",
      "results": [
        {
          "label": "ocr",
          "ocrData": [
            "This topic describes how to call an operation to submit asynchronous image moderation tasks."
          ],
          "ocrLocations": [
            {
              "h": 19,
              "text": "This topic describes how to call an operation to submit asynchronous image moderation tasks.",
              "w": 362,
              "x": 31,
              "y": 11
            }
          ],
          "rate": 99.91,
          "scene": "ocr",
          "suggestion": "review"
        }
      ],
      "taskId": "aaa25f95-4892-4d6b-aca9-7939bc6e9baa-1486198766695",
      "url": "https://test_image_xxxx.png"
    }
  ],
  "msg": "OK",
  "requestId": "992C7849-AA45-4055-8F82-8D44D64C15E3"
}
```

7.Video

7.1. Video moderation

7.1.1. Moderate videos synchronously

This topic describes the `/green/video/syncscan` operation that you can call to moderate videos for risky and illegal content and obtain moderation results in real time. This operation is applicable to the following scenarios: pornography detection, terrorist content detection, ad violation detection, undesirable scene detection, and logo detection.

Scenarios

Scenario	Description	Category of the moderation result
Pornography detection	Detects pornographic content in a video.	normal and porn
Terrorist content detection	Detects terrorist content in a video.	normal and terrorism
Undesirable scene detection	Detects undesirable scenes, such as black or white screen, in a video.	normal and live
Logo detection	Detects specific logos in a video.	normal and logo
Ad violation detection	Detects ads or text violations in a video.	normal and ad
Audio anti-spam	Detects audio violations in a video.	normal, spam, ad, politics, terrorism, abuse, porn, flood, contraband, and customized

Note To use this scenario, you must call the `/green/video/asyncscan` operation to submit asynchronous video moderation tasks. For more information, see [Moderate videos asynchronously](#).

Note By default, the audio to be moderated must be in Chinese. If you need to moderate audio in English, submit a [ticket](#) to contact Alibaba Cloud technical support.

Description

Operation: `/green/video/syncscan`

You can call this operation to submit video moderation tasks and obtain moderation results in real time. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

If you moderate a video in multiple scenarios at a time, you are charged the cumulative fee of all scenarios. The fee of each scenario equals the number of video frames that are moderated in the scenario multiplied by the unit price of the scenario .

- **Moderation object :**

To call this operation to moderate a video and obtain moderation results in real time, you must submit a sequence of frames that are captured from the video to be moderated. If you need to submit a video URL to specify the video to be moderated, we recommend that you call the [/green/video/asyncscan](#) operation to submit asynchronous video moderation tasks.


- **Limits on frames captured from a video:**

- The videos must use HTTP or HTTPS URLs.
- The videos must be in the AVI, FLV, MP4, MPG, ASF, WMV, MOV, WMA, RMVB, RM, FLASH, or TS format.
- A video can be up to 200 MB in size. If you have special requirements, for example, you want to moderate videos larger than 200 MB in size, submit a to raise the threshold.
- The video streams must be transferred by using the RTMP, HLS, HTTP-FLV, or RTSP protocol.
- The duration of a video stream moderation task can be up to 24 hours. If the duration exceeds 24 hours, the video moderation task automatically ends.
- The duration of a video moderation task varies based on the duration for downloading the video. Make sure that you use a stable and reliable storage service to store the videos to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS).

Request parameters

Parameter	Type	Required	Description
bizType	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
scenes	String array	Yes	The video moderation scenario. Valid values: <ul style="list-style-type: none"> • <i>porn</i>: pornography detection • <i>terrorism</i>: terrorist content detection • <i>live</i>: undesirable scene detection • <i>logo</i>: logo detection • <i>ad</i>: ad violation detection
tasks	JSON array	Yes	The list of video moderation tasks. Each element in the JSON array is a structure, namely, each element corresponds to a video. For more information about the structure of each element, see task .

task


Parameter	Type	Required	Description
clientInfo	JSON structure	No	<p>The information about the client. For more information, see the "Common query parameters" section of the Common parameters topic.</p> <p>The server determines whether to use the global clientInfo parameter or the clientInfo parameter that is described in this table.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note The clientInfo parameter in this table takes priority over the global one.</p> </div>
dataId	String	No	The ID of the video to be moderated. Make sure that each ID is unique in a request.
frames	JSON array	Yes	The information about frames that are captured from the video to be moderated. Each element in the JSON array of the frames parameter is a structure, namely, each element corresponds to a captured frame. For more information about the structure of each element, see frame .
framePrefix	String	No	The prefix of the URL of a captured frame, which is used with <code>frame.url</code> to form the complete URL of a captured frame. The complete URL of a captured frame is in the format of <code>framePrefix + frame.url</code> .

frame

Parameter	Type	Required	Description
url	String	Yes	The URL of the captured frame, which is used with <code>framePrefix</code> to form the complete URL of a captured frame. The complete URL of a captured frame is in the format of <code>framePrefix + frame.url</code> .
offset	Integer	No	The interval between the start of the video and the captured frame. Unit: seconds.

Response parameters

Parameter	Type	Description
code	Integer	The HTTP status code returned for the video moderation task.
msg	String	The message returned for the video moderation task.

Parameter	Type	Description
dataId	String	The ID of the moderation object. <div style="background-color: #e0f2f7; padding: 5px;"><p> Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.</p></div>
taskId	String	The ID of the video moderation task.
results	Array	The return results of the video moderation task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result .

result

Parameter	Type	Description
scene	String	The moderation scenario of the moderated video, which you specify in the moderation request. Valid values: <ul style="list-style-type: none">• <i>porn</i>: pornography detection• <i>terrorism</i>: terrorist content detection• <i>live</i>: undesirable scene detection• <i>logo</i>: logo detection• <i>ad</i>: ad violation detection•

Parameter	Type	Description
label	String	<p>The category of the moderation result of the moderated video. Valid values vary based on the specified moderation scenario.</p> <ul style="list-style-type: none"> If the scenes parameter is set to <i>porn</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain pornographic content. <i>porn</i>: The video contains pornographic content. If the scenes parameter is set to <i>terrorism</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain terrorist content. <i>terrorism</i>: The video contains terrorist content. If the scenes parameter is set to <i>live</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain undesirable scenes. <i>live</i>: The video contains an undesirable scene. If the scenes parameter is set to <i>logo</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain logos. <i>logo</i>: The video contains a logo. If the scenes parameter is set to <i>ad</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain ads or text violations. <i>ad</i>: The video contains an ad or a text violation.
sublabel	String	<p>The subcategory of the moderation result. If the scenes parameter is set to <i>porn</i> or <i>terrorism</i>, the subcategory of the moderation result can be returned.</p> <p>By default, this parameter is not returned. If you want this parameter to be returned, submit a ticket.</p>
suggestion	String	<p>The machine-assisted moderation result of the moderated video. Valid values:</p> <ul style="list-style-type: none"> <i>pass</i>: The video does not require further actions. <i>review</i>: The video contains suspected violations and requires human review. <i>block</i>: The video contains violations. We recommend that you delete or block the video.
rate	Floatin g point	<p>The probability that the moderated video falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the video falls into this category.</p>
frames	JSON array	<p>The moderation results of captured frames. For more information about the structure, see frame.</p>

Parameter	Type	Description
extras	JSON object	<p>The additional information about the moderated video.</p> <p>If you specify <i>ad</i> in the <i>scenes</i> parameter to detect ad violations, the <i>extras</i> parameter can return the following content:</p> <p><i>hitLibInfo</i>: the information about the custom text library that the detected ad in the moderated video hits. For more information about the structure, see hitLibInfo.</p> <p>Example:</p> <pre>"hitLibInfo": [{"context": "Haokan", "libCode": "2144002", "libName": "Text pattern library a"}]</pre>
hintWordsInfo	Array	<p>The information about the term that the detected ad in the moderated video hits. For more information about the structure, see hintWordsInfo.</p> <p>Note This parameter is applicable only to ad violation detection.</p> <p>Example:</p> <pre>"hintWordsInfo": [{"context": "Sensitive words"}]</pre>
logoData	Array	<p>The information about the detected logo in the moderated video. For more information about the structure, see logoData.</p> <p>Note This parameter is applicable only to logo detection.</p>
sfaceData	Array	<p>The information about the detected terrorist content in the moderated video. For more information about the structure, see sfaceData.</p> <p>Note This parameter is applicable only to terrorist content detection.</p>

frame

Parameter	Type	Description
url	String	The URL of the captured frame.

Parameter	Type	Description
offset	Integer	The interval between the start of the video and the captured frame. Unit: seconds.
label	String	<p>The category of the moderation result of the captured frame. Valid values vary based on the specified moderation scenario.</p> <ul style="list-style-type: none"> If the scenes parameter is set to <i>porn</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain pornographic content. <i>porn</i>: The video contains pornographic content. If the scenes parameter is set to <i>terrorism</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain terrorist content. <i>terrorism</i>: The video contains terrorist content. If the scenes parameter is set to <i>live</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain undesirable scenes. <i>live</i>: The video contains an undesirable scene. If the scenes parameter is set to <i>logo</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain logos. <i>logo</i>: The video contains a logo. If the scenes parameter is set to <i>ad</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain ads or text violations. <i>ad</i>: The video contains an ad or a text violation.
rate	Floatin g point	The probability that the captured frame falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the captured frame falls into this category.

logoData

Parameter	Type	Description
type	String	The type of the detected logo. For example, a value of TV indicates the logo of a TV station.
name	String	The name of the detected logo.
x	Floatin g point	The distance between the upper-left corner of the logo area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.

Parameter	Type	Description
y	Floatin g point	The distance between the upper-left corner of the logo area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the logo area. Unit: pixels.
h	Floatin g point	The height of the logo area. Unit: pixels.

sfacedata

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the face area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the face area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the face area. Unit: pixels.
h	Floatin g point	The height of the face area. Unit: pixels.
faces	Array	The information about the recognized face in the moderated image. The array contains the following parameters: <ul style="list-style-type: none"> • name: the name of the recognized face that is detected in the moderated image. The value is a string. • rate: the probability that the detected face in the moderated image hits the recognized face. The value is a floating-point number. • id: the ID of the recognized face. The value is a string.

hitLibInfo

Parameter	Type	Description
context	String	The custom text that the detected ad or illegal text in the moderated image hits.
libCode	String	The code of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.
libName	String	The name of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.

hintWordsInfo

Parameter	Type	Description
context	String	The term that the detected ad or illegal text in the moderated image hits.

Examples

Sample requests

```
{
  "scenes": [
    "porn"
  ],
  "tasks": [
    {
      "dataId": "videoid xxx",
      "frames": [
        {
          "offset": 10,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460000"
        },
        {
          "offset": 20,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460001"
        },
        {
          "offset": 30,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460002"
        },
        {
          "offset": 40,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460003"
        },
        {
          "offset": 50,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460003"
        },
        {
          "offset": 60,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A046000x"
        }
      ]
    }
  ]
}
```

Sample success responses

```
{
  "code": 200,
  "msg": "OK",
  "requestId": "requestID xxx",
  "data": [
    {
      "code": 200,
      "msg": "OK",
      "dataId": "videoId xxx",
      "taskId": "taskId xxx",
      "results": [
        {
          "label": "porn",
          "rate": 99.2,
          "scene": "porn",
          "suggestion": "block",
          "frames": [
            {
              "offset": 50,
              "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460003",
              "label": "porn",
              "rate": 99.1
            }
          ]
        }
      ]
    }
  ]
}
```

7.1.2. Moderate videos asynchronously

This topic describes the `/green/video/asyncscan` operation that you can call to asynchronously moderate videos for risky and illegal content. This operation is applicable to the following scenarios: pornography detection, terrorist content detection, ad violation detection, undesirable scene detection, logo detection, and audio anti-spam.

Scenarios

Scenario	Description	Category of the moderation result
Pornography detection	Detects pornographic content in a video.	normal and porn
Terrorist content detection	Detects terrorist content in a video.	normal and terrorism
Undesirable scene detection	Detects undesirable scenes, such as black or white screen, in a video.	normal and live
Logo detection	Detects specific logos in a video.	normal and logo
Ad violation detection	Detects ads or text violations in a video.	normal and ad
Audio anti-spam	<p>Detects audio violations in a video.</p> <p>Note By default, the audio to be moderated must be in Chinese. If you need to moderate audio in English, submit a ticket to contact Alibaba Cloud technical support.</p>	normal, spam, ad, politics, terrorism, abuse, porn, flood, contraband, and customized

Submit asynchronous video moderation tasks

Operation: `/green/video/asyncscan`

You can call this operation to submit asynchronous video moderation tasks. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

You are charged for calling this operation. For more information about the billing method, see [Content Moderation Pricing](#).

If you moderate a video in multiple scenarios at a time, you are charged the cumulative fee of all scenarios. The fee of each scenario equals `the number of video frames that are moderated in the scenario multiplied by the unit price of the scenario`. If you moderate the audio in the video at the same time, you are charged an extra fee for audio anti-spam. The extra fee equals the video duration multiplied by the unit price of audio anti-spam.

- **Moderation object:**

You can call this operation to moderate videos or video streams. You can submit a sequence of frames that are captured from a video or a video URL to specify the video to be moderated.



- **Return results:**

If you send asynchronous moderation requests, the moderation results are not returned in real time. To obtain moderation results, you can poll the moderation results at regular intervals or enable callback notification. The moderation results are retained for up to 1 hour.

- Enable callback notification to obtain video moderation results: When you submit asynchronous video moderation tasks, you can specify a callback URL for receiving moderation results in the callback parameter of the moderation request. For more information about the callback parameter, see [Request parameters](#).
 - Poll video moderation results at regular intervals: You do not need to set the callback parameter when you submit asynchronous video moderation tasks. After you submit the tasks, you can call the `/green/video/results` operation to query moderation results. For more information about the `/green/video/results` operation, see [Query asynchronous video moderation results](#).
- **Limits on videos:**
 - The videos must use HTTP or HTTPS URLs.
 - The videos must be in the AVI, FLV, MP4, MPG, ASF, WMV, MOV, WMA, RMVB, RM, FLASH, or TS format.
 - A video can be up to 200 MB in size. If you have special requirements, for example, you want to moderate videos larger than 200 MB in size, submit a to raise the threshold.
 - The video streams must be transferred by using the RTMP, HLS, HTTP-FLV, or RTSP protocol.
 - The duration of a video stream moderation task can be up to 24 hours. If the duration exceeds 24 hours, the video moderation task automatically ends.
 - The duration of a video moderation task varies based on the duration for downloading the video. Make sure that you use a stable and reliable storage service to store the videos to be moderated. We recommend that you use Alibaba Cloud Object Storage Service (OSS).

Request parameters

Parameter	Type	Required	Description
<code>bizType</code>	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
<code>live</code>	Boolean	No	Specifies whether to moderate live streams. The default value is <i>false</i> , which indicates to moderate videos but not live streams. To moderate live streams, you must set this parameter to <i>true</i> .



Parameter	Type	Required	Description
offline	Boolean	No	<p>Specifies whether to enable the nearline moderation mode.</p> <ul style="list-style-type: none"> The default value is <i>false</i>, which indicates the real-time moderation mode. In this mode, Content Moderation rejects moderation requests that exceed the concurrency limit. If you set this parameter to <i>true</i>, the nearline moderation mode is enabled. In this mode, the moderation tasks you submit may not be processed in real time, but can be queued for processing and will start within 24 hours. <p> Note This parameter is applicable only to video moderation but not video stream moderation.</p>
scenes	String array	Yes	<p>The video moderation scenario. Valid values:</p> <ul style="list-style-type: none"> <i>porn</i>: pornography detection <i>terrorism</i>: terrorist content detection <i>live</i>: undesirable scene detection <i>logo</i>: logo detection <i>ad</i>: ad violation detection
audioScenes	String array	No	<p>The audio moderation scenario. Set the value to <i>antispam</i>.</p> <p>If you do not set this parameter, Content Moderation moderates only the images in videos. If you set this parameter, Content Moderation also moderates the audio in a video during video moderation.</p> <p> Note To moderate the audio in a video, you must set the <i>url</i> parameter in the <i>tasks</i> parameter to submit the URL of the video or video stream. Audio moderation is inapplicable if you set the <i>frames</i> parameter in the <i>tasks</i> parameter to submit frames that are captured from a video.</p>

Parameter	Type	Required	Description
callback	String	No	<p>The callback URL for notifying you of asynchronous moderation results. HTTP and HTTPS URLs are supported. If you do not set this parameter, you must poll moderation results at regular intervals.</p> <p>If you set the callback parameter in the moderation request, make sure that the specified HTTP or HTTPS URL meets the following requirements: supports the POST method, uses UTF-8 to encode the transmitted data, and supports the checksum and content parameters. To send moderation results to the specified callback URL, Content Moderation returns the checksum and content parameters in callback notifications based on the following rules and format:</p> <ul style="list-style-type: none"> checksum: the string in the <code>UID + Seed + Content</code> format that is generated by the Secure Hash Algorithm 256 (SHA-256) algorithm. UID indicates the ID of your Alibaba Cloud account. You can query the ID in the Alibaba Cloud Management Console. To prevent data tampering, you can use the SHA-256 algorithm to generate a string when your server receives a callback notification and verify the string against the received checksum parameter. <p>Note UID cannot be the ID of a RAM user.</p> <ul style="list-style-type: none"> content: the JSON-formatted string to be parsed to the callback data in the JSON format. For more information about the format of the content parameter, see the sample success responses of each operation that you can call to query asynchronous moderation results. <p>Note If your server receives a callback notification, the server sends HTTP status code 200 to Content Moderation. If your server fails to receive a callback notification, the server sends other HTTP status codes to Content Moderation. After Content Moderation receives an HTTP status code other than 200, Content Moderation continues to push the callback notification until your server receives it. Content Moderation can push a callback notification repeatedly for up to 16 times. After 16 times, Content Moderation stops pushing the callback notification. In this case, we recommend that you check the status of the callback URL.</p>

Parameter	Type	Required	Description
seed	String	No	<p>A random string that is used to generate a signature for the callback notification request.</p> <p>The string can be up to 64 characters in length and can contain letters, digits, and underscores (_). You can customize this string. It is used to verify the callback notification request when Content Moderation pushes callback notifications to your server.</p> <p>Note This parameter is required if you set the callback parameter.</p>
tasks	JSON array	Yes	<p>The list of video moderation tasks. Each element in the JSON array is a structure and corresponds to a video. For more information about the structure of each element, see task.</p>

task

Parameter	Type	Required	Description
clientInfo	JSON structure	No	<p>The information about the client. For more information, see the "Common query parameters" section of the Common parameters topic.</p> <p>The server determines whether to use the global clientInfo parameter or the clientInfo parameter that is described in this table.</p> <p>Note The clientInfo parameter in this table takes priority over the global one.</p>
dataId	String	No	<p>The ID of the video to be moderated. Make sure that each ID is unique in a request.</p>
url	String	No	<p>The URL of the video to be moderated. You must set one of the url and frames parameters.</p> <p>Note If you set this parameter, you are charged for the video moderation task based on the video URL that is submitted.</p>


Parameter	Type	Required	Description
frames	JSON array	No	<p>The information about frames that are captured from the video to be moderated. Each element in the JSON array of the frames parameter is a structure and corresponds to a captured frame. For more information about the structure of each element, see frame.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note If you set this parameter, you are charged for the video moderation task based on the captured frames that are submitted.</p> </div>
framePrefix	String	No	<p>The prefix of the URL of a captured frame, which is used with <code>frame.url</code> to form the complete URL of a captured frame. The complete URL of a captured frame is in the format of <code>framePrefix + frame.url</code>.</p>
interval	Integer	No	<p>The interval for capturing frames from the video. Unit: seconds. Valid values: 1 to 60. Default value: 1.</p>
maxFrames	Integer	No	<p>The maximum number of frames that can be captured from the video. Valid values: 5 to 3600. Default value: 200. If you need to capture more frames from the video, submit a ticket.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> This parameter is valid only when you set the live parameter to <code>false</code> to moderate videos but not live streams. If you set the live parameter to <code>true</code>, the number of frames that can be captured from live streams is not limited. If you use an Object Storage Service (OSS) URL that starts with <code>oss://</code> as the source URL of a video and authorize Content Moderation to access ApsaraVideo Media Processing, you can capture a maximum of 20,000 frames from the video. This method does not incur extra charges. For more information, see Authorize a role to access ApsaraVideo for Media Processing. </div>

frame

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

Parameter	Type	Required	Description
url	String	No	The URL of the captured frame, which is used with <code>framePrefix</code> to form the complete URL of a captured frame. The complete URL of a captured frame is in the format of <code>framePrefix + frame.url</code> .
offset	Integer	No	The interval between the start of the video and the captured frame. Unit: seconds.

Response parameters

Parameter	Type	Description
taskId	String	The ID of the video moderation task.
dataId	String	The ID of the moderation object.  Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.

Examples

Sample requests

- Moderate frames that are captured from a video

```
{
  "scenes": [
    "porn"
  ],
  "tasks": [
    {
      "dataId": "videoid xxx",
      "frames": [
        {
          "offset": 10,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460000"
        },
        {
          "offset": 20,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460001"
        },
        {
          "offset": 30,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460002"
        },
        {
          "offset": 40,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460003"
        },
        {
          "offset": 50,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A0460003"
        },
        {
          "offset": 60,
          "url": "http://g1.ykimg.com/0B860000586C0A0300038A046000x"
        }
      ]
    }
  ]
}
```

- Moderate a video

```
{
  "scenes": [
    "porn"
  ],
  "audioScenes": [
    "antispam"
  ],
  "tasks": [
    {
      "dataId": "videoid xxx",
      "url": "http://www.foo.bak/a.mp4",
      "interval": 1,
      "maxFrames": 200
    }
  ]
}
```

- Moderate a live stream

```
{
  "scenes": [
    "porn"
  ],
  "live": true,
  "tasks": [
    {
      "dataId": "videoid xxx",
      "url": "http://www.foo.bak/a.flv",
      "interval": 1,
      "maxFrames": 200
    }
  ]
}
```

Sample success responses

```

{
  "code": 200,
  "msg": "OK",
  "requestId": "requestID xxx",
  "data": [
    {
      "dataId": "videoid xxx",
      "taskId": "taskId xxx"
    }
  ]
}

```

Query asynchronous video moderation results

Operation: /green/video/results

You can call this operation to query asynchronous video moderation results. For more information about how to construct an HTTP request, see [Request structure](#). You can also select an existing HTTP request. For more information, see [SDK overview](#).

- **Billing method:**

This operation is free of charge.


- **Response time:**


We recommend that you query moderation results at least 30s after you send an asynchronous moderation request. Content Moderation retains moderation results for up to 4 hours. If you query moderation results after 4 hours, the results are deleted.

Request parameters

Parameter	Type	Required	Description
body	JSON array	Yes	The list of IDs of asynchronous moderation tasks that you want to query. You can specify up to 1,000 task IDs.

Response parameters




Parameter	Type	Description
code	Integer	The HTTP status code that is returned for the moderation task.
msg	String	The message that is returned for the moderation task.
dataId	String	The ID of the moderation object. <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.</p> </div>

Parameter	Type	Description
taskId	String	The ID of the video moderation task.
results	Array	<p>The return results of the video moderation task. If HTTP status code 200 is returned after a successful call, the array in the return results contains one or more elements. Each element is a structure. For more information about the structure of each element, see result.</p> <div style="background-color: #e0f2f1; padding: 10px; border: 1px solid #ccc;"> <p> Note In the scenario of moderating a video stream, HTTP status code 280 indicates that the moderation task is in progress, and HTTP status code 200 indicates that the moderation task is completed. If the moderation task is in progress, the returned moderation results contain all the issues that Content Moderation has detected in the task.</p> </div>
audioScanResults	Structure	The audio moderation results. For more information about the structure, see audioScanResult .

result


Parameter	Type	Description
scene	String	<p>The moderation scenario of the moderated video, which you specify in the moderation request. Valid values:</p> <ul style="list-style-type: none"> • <i>porn</i>: pornography detection • <i>terrorism</i>: terrorist content detection • <i>live</i>: undesirable scene detection • <i>logo</i>: logo detection • <i>ad</i>: ad violation detection •

Parameter	Type	Description
label	String	<p>The category of the moderation result of the moderated video. Valid values vary based on the specified moderation scenario.</p> <ul style="list-style-type: none"> If the scenes parameter is set to <i>porn</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain pornographic content. <i>porn</i>: The video contains pornographic content. If the scenes parameter is set to <i>terrorism</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain terrorist content. <i>terrorism</i>: The video contains terrorist content. If the scenes parameter is set to <i>live</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain undesirable scenes. <i>live</i>: The video contains an undesirable scene. If the scenes parameter is set to <i>logo</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain logos. <i>logo</i>: The video contains a logo. If the scenes parameter is set to <i>ad</i>, the valid values of the label parameter are: <ul style="list-style-type: none"> <i>normal</i>: The video does not contain ads or text violations. <i>ad</i>: The video contains an ad or a text violation.
sublabel	String	<p>The subcategory of the moderation result. If the scenes parameter is set to <i>porn</i> or <i>terrorism</i>, the subcategory of the moderation result can be returned.</p> <p>By default, this parameter is not returned. If you want this parameter to be returned, submit a ticket.</p>
suggestion	String	<p>The machine-assisted moderation result of the moderated video. Valid values:</p> <ul style="list-style-type: none"> <i>pass</i>: The video does not require further actions. <i>review</i>: The video contains suspected violations and requires human review. <i>block</i>: The video contains violations. We recommend that you delete or block the video.
rate	Floatin g point	<p>The probability that the moderated video falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the video falls into this category.</p>
frames	JSON array	<p>The moderation results of captured frames. For more information about the structure, see frame.</p>

Parameter	Type	Description
extras	JSON object	<p>The additional information about the moderated video.</p> <p>If you specify <i>ad</i> in the scenes parameter to detect ad violations, the extras parameter can return the following content:</p> <p>hitLibInfo: the information about the custom text library that the detected ad in the moderated video hits. For more information about the structure, see hitLibInfo.</p> <p>Example:</p> <pre>"hitLibInfo": [{"context": "Haokan", "libCode": "2144002", "libName": "Text pattern library a"}]</pre>
hintWordsInfo	Array	<p>The information about the term that the detected ad in the moderated video hits. For more information about the structure, see hintWordsInfo.</p> <p> Note This parameter is applicable only to ad violation detection.</p> <p>Example:</p> <pre>"hintWordsInfo": [{"context": "Sensitive words"}]</pre>
logoData	Array	<p>The information about the detected logo in the moderated video. For more information about the structure, see logoData.</p> <p> Note This parameter is applicable only to logo detection.</p>
sfaceData	Array	<p>The information about the detected terrorist content in the moderated video. For more information about the structure, see sfaceData.</p> <p> Note This parameter is applicable only to terrorist content detection.</p>

frame

Parameter	Type	Description
url	String	The URL of the captured frame.
offset	Integer	The interval between the start of the video and the captured frame. Unit: seconds.
		The category of the moderation result of the captured frame. Valid values

Parameter	Type	Description
label	String	<p>vary based on the specified moderation scenario.</p> <ul style="list-style-type: none"> • If the scenes parameter is set to porn, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ normal: The frame does not contain sexy or pornographic content. ◦ sexy: The frame contains sexy content. ◦ porn: The frame contains pornographic content. • If the scenes parameter is set to terrorism, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ normal: The frame does not contain terrorist content. ◦ bloody: The frame contains bloody content. ◦ explosion: The frame contains explosion and smoke. ◦ outfit: The frame contains a special costume. ◦ logo: The frame contains a logo. ◦ weapon: The frame contains a weapon. ◦ politics: The frame contains political content. ◦ violence: The frame contains violence. ◦ crowd: The frame contains a crowd. ◦ parade: The frame contains a parade. ◦ carcrash: The frame contains a car accident. ◦ flag: The frame contains a flag. ◦ location: The frame contains a landmark. ◦ others: The frame contains other specified content. • If the scenes parameter is set to ad, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ normal: The frame does not contain ads or illegal text. ◦ politics: The text contains political content. ◦ porn: The text contains pornographic content. ◦ abuse: The text contains abuse. ◦ terrorism: The text contains terrorist content. ◦ contraband: The text contains prohibited content. ◦ spam: The text contains junk content. ◦ npx: The frame contains an illegal ad. ◦ qrcode: The frame contains a QR code. ◦ programCode: The frame contains an applet code. ◦ ad: The frame contains other ads. <div style="background-color: #e0f2f7; padding: 10px; margin: 10px 0;"> <p> Note By default, only normal and ad can be returned. If you want to use other categories, submit a ticket.</p> </div> <ul style="list-style-type: none"> • If the scenes parameter is set to live, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ normal: The frame does not contain undesirable scenes. ◦ meaningless: The frame is meaningless, such as black or white screen.

Parameter	Type	Description
		<ul style="list-style-type: none"> ◦ PIP: The frame contains a small picture. ◦ smoking: The frame contains smoking content. ◦ driveline: The frame shows live broadcasting in a running vehicle. <ul style="list-style-type: none"> • If the scenes parameter is set to logo, the valid values of the label parameter are: <ul style="list-style-type: none"> ◦ normal: The frame does not contain logos. ◦ TV: The frame contains a TV station logo. ◦ trademark: The frame contains a trademark.
rate	Floatin g point	The probability that the captured frame falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the captured frame falls into this category.

audioScanResult

Parameter	Type	Description
audioScene	String	The audio moderation scenario of the moderated video. The value is fixed to <i>antispam</i> .
label	String	The category of the audio moderation result for the moderated video in the audio moderation scenario. Valid values: <ul style="list-style-type: none"> • <i>normal</i>: The audio does not contain violations. • <i>spam</i>: The audio contains junk content. • <i>ad</i>: The audio contains ads. • <i>politics</i>: The audio contains political content. • <i>terrorism</i>: The audio contains terrorist content. • <i>abuse</i>: The audio contains abuse. • <i>porn</i>: The audio contains pornographic content. • <i>flood</i>: The audio contains excessive junk content. • <i>contraband</i>: The audio contains prohibited content. • <i>customized</i>: The audio contains custom content. For example, the audio contains a custom term.
suggestion	String	The machine-assisted moderation result of the moderated audio. Valid values: <ul style="list-style-type: none"> • <i>pass</i>: The audio does not require further actions. • <i>review</i>: The audio contains suspected violations and requires human review. • <i>block</i>: The audio contains violations. We recommend that you delete or block the audio.
rate	Floatin g point	The probability that the moderated audio falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the audio falls into this category.

Parameter	Type	Description
details	JSON array	The details about text in the moderated audio. The value is a JSON array that contains one or more elements. Each element corresponds to a text entry. For more information, see detail .

detail

Parameter	Type	Description
startTime	Integer	The start time of the text entry in the audio. Unit: seconds.
endTime	Integer	The end time of the text entry in the audio. Unit: seconds.
text	String	The content of the text entry.
label	String	The category of the moderation result of the text entry. Valid values: <ul style="list-style-type: none"> <i>normal</i>: The audio does not contain violations. <i>spam</i>: The audio contains junk content. <i>ad</i>: The audio contains ads. <i>politics</i>: The audio contains political content. <i>terrorism</i>: The audio contains terrorist content. <i>abuse</i>: The audio contains abuse. <i>porn</i>: The audio contains pornographic content. <i>flood</i>: The audio contains excessive junk content. <i>contraband</i>: The audio contains prohibited content. <i>customized</i>: The audio contains custom content. For example, the audio contains a custom term.
keyword	String	The custom term that the text entry hits.
libName	String	The name of the custom text library that contains the custom term hit by the text entry.

logoData

Parameter	Type	Description
type	String	The type of the detected logo. For example, a value of TV indicates the logo of a TV station.
name	String	The name of the detected logo.
x	Floatin g point	The distance between the upper-left corner of the logo area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.

Parameter	Type	Description
y	Floatin g point	The distance between the upper-left corner of the logo area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the logo area. Unit: pixels.
h	Floatin g point	The height of the logo area. Unit: pixels.

sfacedata

Parameter	Type	Description
x	Floatin g point	The distance between the upper-left corner of the face area and the y-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
y	Floatin g point	The distance between the upper-left corner of the face area and the x-axis, with the upper-left corner of the image being the coordinate origin. Unit: pixels.
w	Floatin g point	The width of the face area. Unit: pixels.
h	Floatin g point	The height of the face area. Unit: pixels.
faces	Array	The information about the recognized face in the moderated image. The array contains the following parameters: <ul style="list-style-type: none"> • name: the name of the recognized face that is detected in the moderated image. The value is a string. • rate: the probability that the detected face in the moderated image hits the recognized face. The value is a floating-point number. • id: the ID of the recognized face. The value is a string.

hitLibInfo

Parameter	Type	Description
context	String	The custom text that the detected ad or illegal text in the moderated image hits.
libCode	String	The code of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.
libName	String	The name of the library that contains the custom text hit by the detected ad or illegal text in the moderated image.

hintWordsInfo

Parameter	Type	Description
context	String	The term that the detected ad or illegal text in the moderated image hits.

Examples

Sample requests

```
[
  "taskId xxx",
  "taskId bbb"
]
```

Sample success responses

- Detect only the images in a video

```
{
  "code": 200,
  "msg": "OK",
  "requestId": "requestID xxx",
  "data": [
    {
      "code": 200,
      "msg": "OK",
      "dataId": "videoId xxx",
      "taskId": "taskId xxx",
      "results": [
        {
          "label": "porn",
          "rate": 99.2,
          "scene": "porn",
          "suggestion": "block"
        }
      ]
    }
  ]
}
```

- Detect both the images and audio in a video

```
{
  "code": 200,
  "msg": "OK",
  "requestId": "requestID xxx".
}
```

```
    "data": [
      {
        "code": 200,
        "msg": "OK",
        "dataId": "videoId xxx",
        "taskId": "taskId xxx",
        "results": [
          {
            "label": "porn",
            "rate": 99.2,
            "scene": "porn",
            "suggestion": "block"
          }
        ],
        "audioScanResults": [
          {
            "audioScene": "antispam",
            "label": "customized",
            "suggestion": "block",
            "rate": 99.91,
            "details": [
              {
                "startTime": 0,
                "endTime": 24,
                "text": "blabla...",
                "label": "customized"
              },
              {
                "startTime": 24,
                "endTime": 60,
                "text": "blabla...",
                "label": "normal"
              }
            ]
          }
        ]
      }
    ]
  }
}
```


7.1.3. Stop moderation tasks


This topic describes how to call an operation to stop video moderation tasks. It intends to help you construct an HTTP call request.

- For more information about how to construct an HTTP request, see [Request structure](#).
- You can also select an existing HTTP request. For more information, see [SDK reference](#).

Description

Operation: /green/video/cancelscan

You can call this operation to stop video moderation tasks being performed by the /green/video/asyncscan operation at any time.

 **Note** You are not charged for calling this operation. However, you can call this operation to stop moderating only video streams, but not videos.

Request parameters

For more information about the common request parameters that must be included in all Content Moderation API requests, see [Common request parameters](#).

The request body is a JSON object. The following table describes the parameter that the JSON object contains.

Parameter	Type	Required	Description
body	JSON array	Yes	The list of IDs of moderation tasks you want to stop. You can specify up to 100 task IDs.

Response parameters

The data parameter in the response body is a JSON array. The following table describes the parameters that each element of the JSON array contains.

Parameter	Type	Required	Description
code	Integer	Yes	The HTTP status code returned for the moderation task. A value of 200 indicates a successful call.
msg	String	Yes	The message returned for the moderation task.
dataId	String	No	The ID of the moderated video, which you specify in the dataId parameter of the moderation request.
taskId	String	Yes	The ID returned by the Content Moderation server, which uniquely identifies the moderation task.

Examples

Sample requests

```
[  
  "taskId1",  
  "taskId2"  
]
```

Sample success responses

```
{  
  "code":200,  
  "msg":"OK",  
  "requestId":"AD87F1D1-A675-4F82-8D67-ED43B6FFA876",  
  "data":[  
    {  
      "code":200,  
      "msg":"ok",  
      "dataId":"some dataId 1",  
      "taskId":"taskId1"  
    },  
    {  
      "code":200,  
      "msg":"ok",  
      "dataId":"some dataId 2",  
      "taskId":"taskId2"  
    }  
  ]  
}
```

7.1.4. Give feedback on moderation results

If you find that video moderation results returned by Alibaba Cloud Content Moderation are incorrect, you can call an operation to provide feedback. Alibaba Cloud Content Moderation will optimize the algorithm model based on your feedback.

Description

Operation: /green/video/feedback


You can call this operation to give feedback on the result of a video moderation task and provide your expected result.

 **Note** You are not charged for calling this operation.

Request parameters

For more information about the common request parameters that must be included in all Content Moderation API requests, see [Common request parameters](#).

The request body is a JSON object. The following table describes the parameters that the JSON object contains.

Parameter	Type	Required	Description
taskId	String	Yes	The ID returned by the Content Moderation server, which uniquely identifies the moderation task.
dataId	String	No	The ID of the moderated video, which you specify in the dataId parameter of the moderation request.
url	String	No	The URL of the moderated video, which you specify in the moderation request. If you have not specified the video URL in the moderation request, do not specify this parameter here.
suggestion	String	No	The moderation result you expect to return. Valid values: <ul style="list-style-type: none"> <i>pass</i>: The video is normal. <i>block</i>: The video contains violations.
scenes	String array	No	<p>The scenario of video moderation. Valid values:</p> <ul style="list-style-type: none"> <i>porn</i>: pornography detection <i>terrorism</i>: terrorist content detection <i>ad</i>: ad violation detection <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc;"> <p> Note You can specify multiple moderation scenarios. For example, you can specify both porn and terrorism in the <code>scenes</code> parameter to indicate pornography detection and terrorist content detection scenarios.</p> </div>
frames	JSON array	Yes	The information about captured frames that you consider to be in the detected category. Each element in the JSON array of the frames parameter is a structure. For more information about the structure of each element, see frame .

frame

Parameter	Type	Required	Description
url	String	Yes	The URL of the captured frame.

Parameter	Type	Required	Description
offset	Integer	Yes	The interval between the start of the video and the captured frame. Unit: seconds.
note	String	No	The description of the captured frame.

Response parameters

For more information about common response parameters that this operation returns, see [Common response parameters](#).

The data parameter in the response body is empty.

Examples

Sample requests

```
{
  "taskId": "taskId xxx",
  "dataId": "dataId xxx",
  "url": "http://a.b.c/d.mp4",
  "frames": [
    {
      "url": "http://a.b.c/1.jpg",
      "offset": 100
    },
    {
      "url": "http://a.b.c/2.jpg",
      "offset": 200
    }
  ],
  "suggestion": "block",
  "scenes": [
    "ad",
    "terrorism"
  ],
  "note": "blabla"
}
```

Sample success responses

```
{  
  "code": 200,  
  "msg": "OK",  
  "requestId": "EE5A1189-4D7B-4C24-AD78-4C1FAA3E7A0C"  
}
```

8.Text

8.1. Moderate text for violations

This topic describes how to call an operation to submit text moderation tasks. It intends to help you construct an HTTP call request. You can call the `/green/text/scan` operation to moderate text for violations, such as the pornographic content, ads, spam, political content, and abuse.

- For more information about how to construct an HTTP request, see [Request structure](#).
- You can also select an existing HTTP request. For more information, see [SDK overview](#).


Description

Operation: `/green/text/scan`

Request parameters


Parameter	Type	Required	Description
<code>bizType</code>	String	No	The business scenario. You can create a business scenario in the . For more information, see Customize policies for machine-assisted moderation . You can also submit a to ask Alibaba Cloud engineers to help you create a business scenario.
<code>scenes</code>	String array	Yes	The moderation scenario. Set the value to <i>antispam</i> .
<code>tasks</code>	JSON array	Yes	The list of text moderation tasks. It is a JSON array that contains one or more elements. Each element is a structure. You can add up to 100 elements to the array, which means that you can moderate up to 100 text entries at a time. For more information about the structure of each element, see task .

task

Parameter	Type	Required	Description
<code>clientInfo</code>	JSON structure	No	<p>The information about the client. For more information, see the "Common query parameters" section of the Common parameters topic.</p> <p>The server determines whether to use the global <code>clientInfo</code> parameter or the <code>clientInfo</code> parameter that is described in this table.</p> <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note The <code>clientInfo</code> parameter in this table takes priority over the global one.</p> </div>


Parameter	Type	Required	Description
dataId	String	No	The ID of the text to be moderated. Make sure that each ID is unique in a request.
content	String	Yes	The content of the text to be moderated. Each text entry can be up to 10,000 characters in length.

Response parameters

Parameter	Type	Required	Description
code	Integer	Yes	The HTTP status code returned for the text moderation task.
msg	String	Yes	The message returned for the text moderation task.
dataId	String	No	The ID of the moderation object. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p> Note If you set the dataId parameter in the moderation request, the dataId parameter is returned in the response.</p> </div>
taskId	String	Yes	The ID of the text moderation task.
content	String	No	The content of the moderated text, which you specify in the moderation request.
filteredContent	String	No	The text returned if the original text contains one or more terms in the custom term library, with the terms being replaced with asterisks (*).
results	Array	No	The return results of the text moderation task. If HTTP status code 200 is returned, indicating a successful call, the array in the return results may contain one or more elements. Each element is a structure. For more information about the structure of each element, see result .


result

Parameter	Type	Required	Description
scene	String	Yes	The moderation scenario of the moderated text, which you specify in the moderation request.

Parameter	Type	Required	Description
suggestion	String	Yes	The machine-assisted moderation result of the moderated text. Valid values: <ul style="list-style-type: none"> <i>pass</i>: The text is normal. <i>review</i>: The text requires human review. <i>block</i>: The text contains violations and can be deleted or blocked.
label	String	Yes	The category of moderation results for the moderated text in the specified moderation scenario. Valid values: <ul style="list-style-type: none"> <i>normal</i>: The text is normal. <i>spam</i>: The text contains junk content. <i>ad</i>: The text contains ads. <i>politics</i>: The text contains political content. <i>terrorism</i>: The text contains terrorist content. <i>abuse</i>: The text contains abuse. <i>porn</i>: The text contains pornographic content. <i>flood</i>: The text contains spam. <i>contraband</i>: The text contains prohibited content. <i>meaningless</i>: The text contains meaningless content. <i>customized</i>: The text contains custom content. For example, the text contains a custom term.
rate	Floatin g point	Yes	The probability that the moderated text falls into the category indicated by the label parameter. Valid values: 0.00 to 100.00. A larger value indicates a higher probability that the text falls into the category. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note The value is only for reference. Pay attention to the label and suggestion parameters.</p> </div>
extras	JSON object	No	The additional information about the moderated text.
details	Array	No	The details of the risky content that the moderated text hits. A text entry can hit multiple pieces of risky content. For more information about the structure of each piece of risky content, see detail .

detail

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

Parameter	Type	Required	Description
label	String	Yes	<p>The category of the risky content that the moderated text hits in the specified moderation scenario. Valid values:</p> <ul style="list-style-type: none"> • <i>spam</i>: The text contains junk content. • <i>ad</i>: The text contains ads. • <i>politics</i>: The text contains political content. • <i>terrorism</i>: The text contains terrorist content. • <i>abuse</i>: The text contains abuse. • <i>porn</i>: The text contains pornographic content. • <i>flood</i>: The text contains spam. • <i>contraband</i>: The text contains prohibited content. • <i>meaningless</i>: The text contains meaningless content. • <i>customized</i>: The text contains custom content. For example, the text contains a custom term.
contexts	Array	No	<p>The context information of the risky content that the moderated text hits. For more information about the structure of each piece of context information, see context.</p>
hintWords	JSON array	No	<p>The information about the term that the moderated text hits, which is used to indicate the reason for the violation. Multiple terms that the text hits may be returned. For more information about the structure of each term, see hintWord.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note By default, the value of this parameter is not returned. If you need the value, submit a ticket.</p> </div>

context

Parameter	Type	Required	Description
context	String	Yes	<p>The context of the risky content that the moderated text hits. If the text hits a term or text pattern in your custom text library, the term or text pattern is returned.</p>
libName	String	No	<p>The name of the custom text library returned if the moderated text hits a term or text pattern in the library.</p>
libCode	String	No	<p>The code of the custom text library returned if the moderated text hits a term or text pattern in the library.</p>

Parameter	Type	Required	Description
ruleType	String	No	The type of the behavior rule returned if the moderated text hits the behavior rule. Valid values: <ul style="list-style-type: none"> • <i>user_id</i> • <i>ip</i> • <i>umid</i> • <i>content</i> • <i>similar_content</i> • <i>imei</i> • <i>imsi</i>

hintWord

Parameter	Type	Required	Description
context	String	Yes	The term that the moderated text hits.

Examples

Sample requests

```
{
  "scenes": [
    "antispam"
  ],
  "tasks": [
    {
      "dataId": "xxxx$rdBjUC1C-1rd9Ah",
      "content": "Obama met with Trump in the White House yesterday."
    }
  ]
}
```

Sample success responses

```
{
  "msg": "OK",
  "code": 200,
  "data": [
    {
      "msg": "OK",
      "code": 200,
      "dataId": "xxxx$rdBjUC1C-1rd9Ah",
      "results": [
        {
          "rate": 50.0,
          "suggestion": "review",
          "details": [
            {
              "hintWords": [
                {
                  "context": "Obama"
                }
              ],
              "contexts": [
                {
                  "libCode": "123456",
                  "libName": "Name of your custom text library",
                  "context": "Trump"
                }
              ],
              "label": "politics"
            }
          ],
          "label": "politics",
          "scene": "antispam"
        }
      ],
      "content": "Obama met with Trump in the White House yesterday.",
      "filteredContent": "**** met with Trump in the White House yesterday.",
      "taskId": "xxxxxyyyyyy-xxxx"
    }
  ],
  "requestId": "yyyyyyy-862F-4BAE-8B4E-xxxxxx"
}
```

8.2. Give feedback on moderation results

If you find that text moderation results returned by Alibaba Cloud Content Moderation are incorrect, you can call an operation to provide feedback.


Description

Operation: `/green/text/feedback`. You can call this operation to give feedback on the result of a text moderation task.

Request parameters

For more information about the common request parameters that must be included in all Content Moderation API requests, see [Common request parameters](#).

The request body is a JSON array. The following table describes the parameters that each element of the JSON array contains.

Parameter	Type	Required	Description
taskId	String	Yes	The ID returned by the Content Moderation server, which uniquely identifies the moderation task.
dataId	String	No	The ID of the moderated text, which you specify in the dataId parameter of the moderation request.
content	String	No	The content of the moderated text. Each piece of text can be up to 10,000 characters in length.
label	String	No	<p>The expected category of moderation results for the moderated text in the specified moderation scenario. For more information about valid values, see Parameters scenes and label of the <code>/green/text/scan</code> operation.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note If you set the label parameter to normal, the system adds the text to the whitelist of the feedback-based text library in the Alibaba Cloud Content Moderation console. If you set the label parameter to a value other than normal, the system adds the text to the blacklist of the feedback-based text library.</p> </div>
note	String	No	The description of the moderated text, such as the keywords in the text.

Response parameters

For more information about common response parameters that this operation returns, see [Common response parameters](#).

The data parameter in the response body is empty.

Examples

Sample request

```
{
  "dataId": "test7fcmzGkKbNp7@JqPQRnwle-1mwwZX",
  "taskId": "taskid xxx",
  "content": "foobar",
  "label": "spam",
  "note": "blabla ..."
}
```

Sample success responses

```
{
  "msg": "OK",
  "code": 200,
  "requestId": "02C418E7-81D4-4375-9732-900B64D07CEE"
}
```