ALIBABA CLOUD

Alibaba Cloud

内容安全

API Reference (sample library management)

Document Version: 20210525

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 `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Development preparations 05

2.List of operations by function 06

3.Construct a request 08

4.Business scenario management 14

  4.1. Create a business scenario 14

  4.2. Delete a business scenario 15

  4.3. Query business scenarios 16

5.Manage custom image libraries 19

  5.1. Query image libraries 19

  5.2. Create an image library 21

  5.3. Modify the information of an image library 23

  5.4. Delete an image library 24

  5.5. Delete images 25

  5.6. Upload images 26

6.Manage custom text libraries 28

  6.1. Query text libraries 28

  6.2. Create a text library 31

  6.3. Modify a text library 34

  6.4. Delete a text library 35

  6.5. Add text 36

  6.6. Query text 37

  6.7. Delete text 39

1.Development preparations

This topic describes the preparations required for calling the Content Moderation API.

Activate the service

- Register an Alibaba Cloud account on the Alibaba Cloud official website. Skip this step if you have registered an account.
- Open the landing page of Alibaba Cloud Content Moderation and click **Enable Now** to activate the service.

Create an AccessKey

Manage your AccessKey ID and AccessKey secret on the AccessKey management page.

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

Create and authorize a RAM user

Content Moderation allows you to use the AccessKey ID and AccessKey secret of a RAM user to call the Content Moderation API.

Follow these steps to create and authorize a RAM user:

1. Log on to the RAM console, create a RAM user, and then generate its AccessKey ID and AccessKey secret. Properly keep the generated AccessKey ID and AccessKey secret for calling the API operations and SDKs later.

   For more information, see Create a RAM user.

2. Authorize the RAM user. Grant the following system policy permission to the RAM user you created: `AliyunYundunGreenWebFullAccess` .

   For more information, see Grant permissions to a RAM user.

2.List of operations by function

This topic lists all configuration management operations that the Content Moderation API provides. You can call the operations to manage custom text libraries and custom image libraries.

Overview

Content Moderation allows you to manage custom text libraries and custom image libraries. You can complete these tasks in the Alibaba Cloud Content Moderation console or by calling API operations to easily meet your content moderation needs.

> **Note** We recommend that you use the SDKs that Content Moderation provides in different development languages to call the Content Moderation API. The SDKs free you from the complex needs of constructing a request and calculating a signature to call the Content Moderation API. If no SDK is available for your development language, you can refer to the API reference to directly call API operations. For more information about the SDK operations, see SDK overview.

The following table lists the types of configuration management operations that the Content Moderation API provides.

Type	Feature
Operations for managing custom image libraries	Manage custom image libraries in the Alibaba Cloud Content Moderation console.
Operations for managing custom text libraries	Manage custom text libraries in the Alibaba Cloud Content Moderation console.

Operations for managing custom image libraries

Operation	Feature	Description
DescribeImageLib	Queries image libraries.	You can call this operation to query the custom image libraries that have been created.
CreateImageLib	Creates an image library.	You can call this operation to create a custom image library.
UpdateImageLib	Modifies the information of an image library.	You can call this operation to modify the name and BizTypes parameter of a specified custom image library.
DeleteImageLib	Deletes an image library.	You can call this operation to delete a specified custom image library.
DescribeImageFromLib	Queries images in an image library.	You can call this operation to query the images that have been added to a specified custom image library.

Operation	Feature	Description
DeleteImageFromLib	Deletes images.	You can call this operation to delete one or more images from a specified custom image library.
UploadImageToLib	Uploads images.	You can call this operation to add the metadata of images to a specified custom image library.

Operations for managing custom text libraries

Operation	Feature	Description
DescribeKeywordLib	Queries text libraries.	You can call this operation to query the text libraries that have been created.
CreateKeywordLib	Creates a text library.	You can call this operation to create a custom text library.
UpdateKeywordLib	Modifies the basic information of a text library.	You can call this operation to modify the basic information of a specified custom text library.
DeleteKeywordLib	Deletes a text library.	You can call this operation to delete a specified custom text library.
CreateKeyword	Adds text.	You can call this operation to add text to a specified custom text library.
DescribeKeyword	Queries text.	You can call this operation to query text in a specified custom text library.
DeleteKeyword	Deletes text.	You can call this operation to delete text from a specified custom text library.

3.Construct a request

This topic describes the method for constructing a request to call the Content Moderation API. This method is applicable if you initiate HTTP or HTTPS requests based on the API URL.

> **Note** Content Moderation provides SDKs in multiple programming languages and supports third-party SDKs. You can use the SDKs to eliminate the need of constructing API requests by yourself. For more information, see SDK overview.

Protocols

You can send requests over HTTP or HTTPS.

Methods

All operations support the POST and GET methods.

Encoding

The request parameters and responses of all operations are encoded in UTF-8.

API request structure

Type	Description	Remarks
API endpoint	The endpoint for the Content Moderation API.	Example: `http(https)://green.cn-shanghai.aliyuncs.com` .
Common request parameters	The common parameters required in all API requests.	For more information, see Common request parameters.
Operation-specific parameters	The parameters that are specific to each operation.	For more information, see the definition of each operation.

API endpoint

To call the Content Moderation API to manage custom term libraries, custom text pattern libraries, and custom image libraries, you must send a request to the following endpoint:

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

> **Notice** Only the preceding endpoint is supported. Use the preceding endpoint to call the Content Moderation API in all regions of China. The server can automatically synchronize data if you initiate a request in a region other than cn-shanghai.

Common request parameters

Parameter	Type	Required	Description

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. For more information, see List of operations by function.
AccessKeyId	String	Yes	The AccessKey ID provided to you by Alibaba Cloud. The AccessKey is used to call API operations.
Signature	String	Yes	The signature string of the current request. For more information about how signatures are calculated, see Signature method.
SignatureMethod	String	Yes	The encryption method of the signature string. Set the value to *HMAC-SHA1*.
SignatureVersion	String	Yes	The version of the signature encryption algorithm. Set the value to *1.0*.
SignatureNonce	String	Yes	A unique, random number used to prevent replay attacks. You must use different numbers for different requests.
Timestamp	String	Yes	The timestamp of the request. Specify the time in the ISO 8601 standard in the *yyyy-MM-ddTHH:mm:ssZ* format. The time must be in UTC. Example: *2018-01-01T12:00:00Z*.
Version	String	Yes	The version number of the API. The value must be in the *YYYY-MM-DD* format. Set the value to *2017-08-23*.
Format	String	No	The format in which to return the response. Valid values: • *json* • *xml* Default value: xml.

Operation-specific parameters

Click an operation in List of operations by function to query the specific parameters of the operation.

Sign signatures

You must sign all HTTP or HTTPS API requests to gurantee security. Alibaba Cloud uses the request signature to verify the identity of the request sender. Content Moderation implements symmetric encryption with an AccessKey pair to verify the identity of the request sender.

An AccessKey pair is an identity credential issued to Alibaba Cloud accounts and RAM users that is similar to a logon username and password. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is used to verify the identity of the user, while the AccessKey secret is used to encrypt and verify the signature string. You must keep your AccessKey secret strictly confidential.

1. Create a canonicalized query string.

i. Arrange the request parameters (including all common and operation-specific parameters except Signature) in alphabetical order.

> **Note** When you use the GET method to submit requests, specify these parameters in the request URL following the question mark `(?)` and connect the remaining parameters with ampersands `(&)`.

ii. Encode the request parameters and their values in UTF-8 based on RFC 3986. Encoding rules are as follows:

- Uppercase letters, lowercase letters, digits, and some special characters such as hyphens (`-`), underscores (`_`), periods (`.`), and tildes (`~`) do not need to be encoded.
- Other characters must be percent encoded in `%XY` format. `XY` represents the ASCII code of the characters in hexadecimal notation. For example, double quotation marks (`"`) are encoded as `%22`.
- Extended UTF-8 characters are encoded in `%XY%ZA...` format.
- Spaces must be encoded as `%20`. Do not encode spaces as plus signs (`+`).

  The preceding encoding method is similar to but slightly different from the `application/x-www-form-urlencoded` MIME-type encoding algorithm. If you use `java.net.URLEncoder` in the Java standard library, use `percentEncode` to encode request parameters and their values. In the encoded query string, replace the plus sign (`+`) with `%20`, the asterisk (`*`) with `%2A`, and `%7E` with a tilde (`~`). In this way, you can obtain an encoded string that matches the preceding encoding rules.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
return value ! = null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%
2A").replace("%7E", "~") : null;
}
```

iii. Use an equal sign (`=`) to connect each encoded request parameter and its value.

iv. Use an ampersand (`&`) to connect the encoded request parameters. Note that these parameters must be arranged in the same order as that in Step a.

Now, you have obtained a canoncalized query string (CanonicalizedQueryString) that follows the API request structure.

2. Create a string-to-sign from the encoded canonicalized query string.

   i. Create `StringToSign`. You can also use `percentEncode` to encode the canonicalized query string constructed in the previous step. The rules are as follows:

```
StringToSign=
  HTTPMethod + "&" + // HTTPMethod: the HTTP method used to send a request, such as GET.
 percentEncode("/") + "&" + // percentEncode("/"): the UTF-8 encoded value of the forward slash (/)
, which is %2F.
 percentEncode(CanonicalizedQueryString) // Your canonicalized query string.
```

ii. Calculate the HMAC-SHA1 value of `StringToSign` based on RFC 2104. This example uses the Java Base64 encoding method.

```
Signature = Base64( HMAC-SHA1( AccessSecret, UTF-8-Encoding-Of(StringToSign) ) )
```

> **Note** When you calculate the signature, the key value specified by RFC 2104 is your `AccessKeySecret` with an ampersand (`&`) which has an ASCII value of 38.

iii. Encode the `Signature` parameter based on RFC 3986 and add it to the canonicalized query string.

Sample code

- Example 1: Concatenate parameters

  The following example describes how to call the DescribeKeywordLib operation to query custom text libraries. In the case of `AccessKeyID=testid` and `AccessKeySecret=testsecret` , the signature process is as follows:

  i. Create a canonicalized query string.

  ```
  AccessKeyId=testid&Action=DescribeKeywordLib&Format=XML&SignatureMethod=HMAC-SHA1&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&SignatureVersion=1.0&Timestamp=2016-02-23T12:46:24Z&Version=2014-05-26&ServiceModule=open_api
  ```

  ii. Create a string-to-sign.

  ```
  GET&%2F&AccessKeyId%3Dtestid%26Action%3DDescribeKeywordLib%26Format%3DXML%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf%26SignatureVersion%3D1.0%26Timestamp%3D2016-02-23T12%253A46%253A24Z%26Version%3D2014-05-26%26ServiceModule%3Dopen_api
  ```

  iii. Calculate the HMAC value of the string-to-sign. In the case of `AccessKeySecret=testsecret` , the key value used for calculation is `testsecret&` . The calculated signature value is `OLeaidS1JvxuMvnyHOwuJ+uX5qY=` . This example uses the Java Base64 encoding method.

  ```
  Signature = Base64( HMAC-SHA1( AccessSecret, UTF-8-Encoding-Of(StringToSign) ) )
  ```

  iv. Add the `Signature=OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D` string that has been encoded based on RFC 3986 to the URL in Step a.

  ```
  http://green.cn-shanghai.aliyuncs.com/?SignatureVersion=1.0&Action=DescribeKeywordLib&Format=XML&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&AccessKeyId=testid&Signature=OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D&SignatureMethod=HMAC-SHA1&Timestamp=2016-02-23T12%253A46%253A24Z&ServiceModule=open_api
  ```

  You can use a browser or tools such as cURL and Wget to initiate an HTTP request through the new URL. The HTTP request calls the DescribeKeywordLib operation to query custom text libraries.

- Example 2: Use the standard library of the programming language

  The following example describes how to call the DescribeKeywordLib operation to query custom text libraries. Assume that `AccessKeyID` is set to testid and `AccessKeySecret` is set to testsecret, and all request parameters are placed in a Java `Map<String, String>` object.

  i. Predefine the encoding method.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
  return value ! = null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A"
).replace("%7E", "~") : null;
}
```

ii. Predefine the time format for the Timestamp parameter. The value of the Timestamp parameter must be specified in the ISO 8601 standard. The time must be in UTC+0.

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
  SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
  df.setTimeZone(new SimpleTimeZone(0, "GMT"));
  return df.format(date);
}
```

iii. Create a query string.

```
final String HTTP_METHOD = "GET";
Map parameters = new HashMap();
// Specify request parameters.
parameters.put("Action", "DescribeKeywordLib");
parameters.put("Version", "2017-08-23");
parameters.put("AccessKeyId", "testid");
parameters.put("Timestamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1.0");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "XML");
parameters.put("ServiceModule", "open_api");
// Arrange request parameters.
String[] sortedKeys = parameters.keySet().toArray(new String[]{});
Arrays.sort(sortedKeys);
final String SEPARATOR = "&";
// Create a string-to-sign.
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/")).append(SEPARATOR);
StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
// Encode the key and value.
  canonicalizedQueryString.append("&")
  .append(percentEncode(key)).append("=")
  .append(percentEncode(parameters.get(key)));
}
// Encode the canonicalized query string.
stringToSign.append(percentEncode(
  canonicalizedQueryString.toString().substring(1)));
```

iv. Calculate the signature. In the case of `AccessKeySecret=testsecret`, the key value used for calculation is `testsecret&`. The calculated signature value is `OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D`.

```
// The following code demonstrates how to calculate the signature:
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";
Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
String signature = new String(Base64.encodeBase64(signData));
```

Encode the Signature parameter based on RFC 3986 and add the parameter to the URL. The new URL is as follows:

```
http://green.cn-shanghai.aliyuncs.com/?SignatureVersion=1.0&Action=DescribeKeywordLib&Format=XML&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&AccessKeyId=testid&Signature=OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D&SignatureMethod=HMAC-SHA1&Timestamp=2016-02-23T12%253A46%253A24Z
```

v. Use browsers or tools such as cURL and Wget to send HTTP requests and obtain the response.

API response

Click an operation in List of operations by function to query the response of the operation.

4.Business scenario management

4.1. Create a business scenario

This topic describes the CreateBizType operation that you can call to create a business scenario. You can use the business scenario to customize a policy for machine-assisted moderation for the Content Moderation API.

Operation description

Operation: CreateBizType

You can call this operation to create a business scenario. 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.

> ? **Note** By default, you can have up to 100 custom business scenarios. If you need more business scenarios, submit a ticket.

Request parameters

Parameter	Type	Required	Example	Description
BizTypeName	String	Yes	bizType1	The name of the business scenario. The name can contain letters, underscores (_), and digits. The name cannot contain Chinese characters.
BizTypeImport	String	No	bizTypeTest	The name of an existing business scenario to be imported.
CiteTemplate	Boolean	No	true	Specifies whether to import the configuration of an industry template. Valid values: • **true**: imports the configuration of an industry template. • **false**: does not import the configuration of an industry template. Default value: false. ? **Note** If you set this parameter to **true**, you must specify the IndustryInfo parameter.

Parameter	Type	Required	Example	Description
IndustryInfo	String	No	Social-Registration information-Profile picture	The industry classification. Valid values: - Social-Registration information-Profile picture - Social-Registration information-Nickname
Description	String	No	Test business scenario	The description of the business scenario.

Response parameters

Parameter	Type	Example	Description
Message	String	OK	The message that is returned for the request.
RequestId	String	795D8871-4889-4C0F-A8B1-C7D2B990FF61	The ID of the request.

Examples

Sample requests

```
{
 "BizTypeName": "bizType1",
 "BizTypeImport": "bizTypeTest",
 "CiteTemplate": true,
 "IndustryInfo": "Social-Registration information-Profile picture",
 "Description": "Test business scenario"
}
```

Sample success responses

```
{
 "RequestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61"
}
```

4.2. Delete a business scenario

This topic describes the DeleteBizType operation that you can call to delete a business scenario. If you need to delete a business scenario, make sure that the business scenario is not in use. If you use a deleted business scenario, the moderation policy that is customized based on the business scenario becomes unavailable and the default moderation policy is used. This may lead to errors in the moderation of business data.

Operation description

Operation: DeleteBizType

You can call this operation to delete a business scenario. 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.

Request parameters

Parameter	Type	Required	Example	Description
BizTypeName	String	Yes	bizType1	The name of the business scenario.

Response parameters

Parameter	Type	Example	Description
Message	String	OK	The message that is returned for the request.
RequestId	String	795D8871-4889-4C0F-A8B1-C7D2B990FF61	The ID of the request.

Examples

Sample requests

```
{
 "BizTypeName": "bizType1"
}
```

Sample success responses

```
{
 "RequestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
}
```

4.3. Query business scenarios

This topic describes the DescribeUserBizTypes operation that you can call to query existing and custom business scenarios. This helps you manage business scenarios.

Operation description

Operation: DescribeUserBizTypes

You can call this operation to query business scenarios. 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.

Request parameters

None

Response parameters

Parameter	Type	Example	Description
BizTypeList	JSONArray	N/A	The list of all business scenarios. Each element in the JSON array is a structure. For more information about the structure, see BizType.
BizTypeListImport	JSONArray	N/A	The list of business scenarios whose configurations can be imported. Each element in the JSON array is a structure. For more information about the structure, see BizType.
Message	String	OK	The message that is returned for the request.
RequestId	String	ADDBDECA-E1B2-4988-AE3B-956BF51573EC	The ID of the request.

BizType

Parameter	Type	Example	Description
BizType	String	mohongtest2	The name of the business scenario.
CiteTemplate	Boolean	false	Indicates whether an industry template was imported. Valid values: - `true`: An industry template was imported. - `false`: No industry template was imported.
IndustryInfo	String	Social-Registration information-Profile picture	The industry information.
Source	Float	custom	The source of the business scenario. Valid values: - `custom`: a custom business scenario. - `system`: the default business scenario of Content Moderation.
Description	String	Test business scenario	The description of the business scenario.

Examples

Sample success responses

```
{
  "BizTypeList":[
    {
      "CiteTemplate":false,
      "BizType":"mohongtest2",
      "Source":"custom",
      "IndustryInfo":"",
   "Description": ""
    }
  ],
  "RequestId":"ADDBDECA-E1B2-4988-AE3B-956BF51573EC",
  "BizTypeListImport":[
    {
      "CiteTemplate":false,
      "BizType":"mohongtest2",
      "Source":"custom",
      "IndustryInfo":"",
   "Description": ""
    }
  ]
}
```

5.Manage custom image libraries

5.1. Query image libraries

This topic describes how to call the DescribeImageLib operation to query custom image libraries.

Description

Operation: DescribeImageLib

You can call this operation to query custom image libraries.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
ServiceModule	String	Yes	The name of the service module to which the image library is applied. Valid value: *open_api* (indicates the Content Moderation API).

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

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the image library.
ModifiedTime	String	Yes	The last time when the image library was modified.
Name	String	Yes	The name of the image library.
Code	String	Yes	The code of the image library. **Note** The code of the image library appears in the Alibaba Cloud Content Moderation console. The primary key ID is used to identify the image library in the operation.
ImageCount	String	Yes	The number of images in the image library.

Parameter	Type	Required	Description
Category	String	Yes	The category of the image library. Valid values: • *BLACK*: a blacklist. • *WHITE*: a whitelist.
Source	String	Yes	The source of the image library. Valid values: • *MANUAL*: indicates that the image library is manually created. • *FEEDBACK*: indicates that the image library is automatically created based on user feedback.
ServiceModule	String	Yes	The name of the service module to which the image library is applied. Valid value: *open_api* (indicates the Content Moderation API).
Scene	String	Yes	The moderation scenario to which the custom image library applies. Valid values: • *PORN*: pornography detection. • *AD*: ad violation detection. • *ILLEGAL*: terrorist content detection.
BizTypes	List	No	The business type of the custom image library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
Enable	Boolean	No	Indicates whether the image library was enabled. A value of true indicates that the image library was enabled, and a value of false indicates that the image library was disabled.

Examples

Sample requests

```
{
  "ServiceModule": open_api
}
```

Sample success responses

```
{
 "requestId": "6CF2815C-C8C7-4A01-B52E-FF6E24F53492",
 "data": {
  "TotalCount": 5,
  "ImageLibList": [
   {
    "Name": "Test a",
    "Source": "MANUAL",
    "ImageCount": 191231231230,
    "BizTypes": [
     1,
     2,
     3
    ],
    "Category": "BLACK",
    "ServiceModule": "open_api",
    "Scene": "PORN",
    "ModifiedTime": "2018-03-19 10:45:44 +0800",
    "Id": 1279,
    "Code": "808003055",
    "Enable": true
   },
   {
    "Name": "Hello aa",
    "Source": "MANUAL",
    "ImageCount": 0,
    "BizTypes": [],
    "Category": "BLACK",
    "ServiceModule": "open_api",
    "Scene": "PORN",
    "ModifiedTime": "2018-03-01 19:20:37 +0800",
    "Id": 1222,
    "Code": "808003043",
    "Enable": true
   }
  ]
 },
 "code": 200,
 "success": true
}
```

5.2. Create an image library

This topic describes how to call the CreateImageLib operation to create an image library.

Description

Operation: CreateImageLib

You can call this operation to create a custom similar image library.

Note The maximum number of image libraries equals the number of image libraries created by using this operation plus the number of image libraries created in the Alibaba Cloud Content Moderation console.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Name	String	Yes	The name of the image library.
Category	String	Yes	The category of the image library. Valid values: - *BLACK*: a blacklist. - *WHITE*: a whitelist.
ServiceModule	String	Yes	The name of the service module to which the image library is applied. Valid value: *open_api* (indicates the Content Moderation API).
Scene	String	Yes	The moderation scenario to which the custom image library applies. Valid values: - *PORN*: pornography detection. - *AD*: ad violation detection. - *ILLEGAL*: terrorist content detection.
BizTypes	List	No	The business type of the custom image library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
Enable	Boolean	No	Specifies whether to enable the image library. A value of true indicates that the image library is enabled, and a value of false indicates that the image library is disabled.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Name": "test1sdkdk",
 "BizTypes": [],
 "Category": "BLACK",
 "ServiceModule": "open_api",
 "ResourceType": "IMAGE",
 "Scene": "PORN"
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

5.3. Modify the information of an image library

This topic describes how to call the UpdateImageLib operation to modify the basic information of a specified image library.

Description

Operation: UpdateImageLib

You can call this operation to modify the basic information of a specified image library.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the image library.
Name	String	Yes	The name of the image library.
Category	String	Yes	The category of the image library. Valid values: ● *BLACK*: a blacklist. ● *WHITE*: a whitelist.

Parameter	Type	Required	Description
Scene	String	Yes	The moderation scenario to which the custom image library applies. Valid values: • *PORN*: pornography detection. • *AD*: ad violation detection. • *ILLEGAL*: terrorist content detection.
BizTypes	List	No	The business type of the custom image library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
Enable	Boolean	No	Specifies whether to enable the image library. A value of true indicates that the image library is enabled, and a value of false indicates that the image library is disabled.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Id": 2147,
 "Name": "test2sdkdk",
 "BizTypes": []
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

5.4. Delete an image library

This topic describes how to call the DeleteImageLib operation to delete a specified image library.

Description

Operation: DeleteImageLib

You can call this operation to delete a specified image library.

Note Deleted image libraries cannot be restored. Proceed with caution.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the image library to delete. **Note** Do not set this parameter to the code of the image library.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
  "Id": 2147
}
```

Sample success responses

```
{
  "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
  "code": 200
}
```

5.5. Delete images

This topic describes how to call the DeleteImageFromLib operation to delete one or more images from a specified image library.

Description

Operation: DeleteImageFromLib

You can call this operation to delete one or more images from a specified image library.

Note Deleted images cannot be restored. Proceed with caution.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Ids	List	Yes	The IDs of the images to delete.
ImageLibId	String	Yes	The primary key ID of the image library.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Id": [1,2],
 "ImageLibId": 2147
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

5.6. Upload images

This topic describes how to call the UploadImageToLib operation to upload images to a specified image library. Before uploading images, you must call the DescribeUploadInfo operation to obtain a signature for uploading images.

Description

Operation: UploadImageToLib

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
ImageLibId	Numeric	Yes	The primary key ID of the image library.
Urls	List	Yes	The images. Each element in the list is the access URL of an image in OSS.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
  "ImageLibId": "2147",
  "Urls": "[\"http://oss-example.com/1.jpg\"]"
}
```

Sample success responses

```
{
 "RequestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

6.Manage custom text libraries

6.1. Query text libraries

This topic describes how to call the DescribeKeywordLib operation to query the custom text libraries that have been created. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: DescribeKeywordLib

You can call this operation to query custom text libraries.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
ServiceModule	String	Yes	The name of the service module to which the text library is applied. Valid value: *open_api* (indicates the Content Moderation API).

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

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the text library.
ModifiedTime	String	Yes	The last time when the text library was modified.
Name	String	Yes	The name of the text library.
Code	String	Yes	The code of the text library. **Note** The code of the text library appears in the Alibaba Cloud Content Moderation console. The primary key ID is used to identify the text library in the operation.
Count	String	Yes	The number of terms in the text library.

Parameter	Type	Required	Description
Category	String	Yes	The category of the text library. Valid values: - *BLACK*: a blacklist. - *WHITE*: a whitelist. - *REVIEW*: a review list.
Source	String	Yes	The source of the text library. **Note** The value is fixed to *MANUAL*, which indicates that the text library is manually added.
ServiceModule	String	Yes	The name of the service module to which the text library is applied. Valid value: *open_api* (indicates the Content Moderation API).
BizTypes	List	No	The business type of the custom text library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
ResourceType	String	Yes	The resource type of the text library. Valid values: - *TEXT*: a literal text library. - *IMAGE*: a text library against which text extracted from images is matched. - *VOICE*: a text library against which text converted from audio is matched.

LibType	String	Yes	The type of the text library in each moderation scenario. Valid values:
			- **Text anti-spam**
			  - *textKeyword*: a text library against which terms in text are matched.
			  - *similarText*: a text library against which text patterns are matched.
			- **Detection of ads in images**
			textKeyword: a text library against which terms extracted from images are matched.
			- **Audio anti-spam**
			voiceText: a text library against which terms converted from audio are matched.

MatchMode	String	No	The matching method. Valid values: - *fuzzy*: fuzzy matching. - *precise*: exact matching.

Parameter	Type	Required	Description
Enable	Boolean	No	Indicates whether the text library was enabled. A value of true indicates that the text library was enabled, and a value of false indicates that the text library was disabled.

Examples

Sample requests

```
{
  "ServiceModule": open_api
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "data": {
  "TotalCount": 11,
  "KeywordLibList": [
   {
    "Name": "test1sdkdk",
    "Source": "MANUAL",
    "BizTypes": [],
    "Category": "BLACK",
    "ServiceModule": "open_api",
    "Count": 3,
    "ResourceType": "TEXT",
    "Id": 2147,
    "ModifiedTime": "2018-09-04 14:17:56 +0800",
    "Code": "710001",
    "LibType": "textKeyword",
    "MatchMode":"fuzzy",
    "Enable":"true"
   },
   {
    "Name": "Audio test",
    "Source": "MANUAL",
    "BizTypes": [],
    "Category": "BLACK",
    "ServiceModule": "open_api",
    "Count": 3,
    "ResourceType": "VOICE",
    "Id": 2693,
    "ModifiedTime": "2018-09-03 17:11:04 +0800",
    "Code": "932001",
    "LibType": "voiceText",
    "MatchMode":"fuzzy",
    "Enable":"true"
   }
  ]
 },
 "code": 200,
 "success": true
}
```

6.2. Create a text library

This topic describes how to call the CreateKeywordLib operation to create a custom text library. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: CreateKeywordLib

You can call this operation to create a text library. The created text library can be used for text anti-spam.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
ServiceModule	String	Yes	The name of the service module to which the text library is applied. Valid value: *open_api* (indicates the Content Moderation API).
Name	String	Yes	The name of the text library.
ResourceType	String	Yes	The resource type of the text library. Valid values: • *TEXT*: a literal text library. • *IMAGE*: a text library against which text extracted from images is matched. • *VOICE*: a text library against which text converted from audio is matched.
BizTypes	List	No	The business type of the custom text library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
Category	String	Yes	The category of the text library. Valid values: • *BLACK*: a blacklist. • *WHITE*: a whitelist. • *REVIEW*: a review list.
LibType	String	Yes	The type of the text library in each moderation scenario. Valid values: • **Text anti-spam** ◦ *textKeyword*: a text library against which terms in text are matched. ◦ *similarText*: a text library against which text patterns are matched. • **Detection of ads in images** *textKeyword*: a text library against which terms extracted from images are matched. • **Audio anti-spam** *voiceText*: a text library against which terms converted from audio are matched.

Parameter	Type	Required	Description
MatchMode	String	No	The matching method. Valid values: • *fuzzy*: fuzzy matching. • *precise*: exact matching.
Enable	Boolean	No	Specifies whether to enable the text library. A value of true indicates that the text library is enabled, and a value of false indicates that the text library is disabled.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Parameter	Type	Required	Description
code	Integer	Yes	The error code. Most error codes are consistent with HTTP status codes, and some error codes are extended. • 2xx: indicates a success. • 4xx: indicates an incorrect request. • 5xx: indicates a back-end error. For more information, see Common errors.
msg	String	No	The description of the call error.
requestId	String	Yes	The unique ID of the request, which can be used to locate issues.
id	String	Yes	The unique ID of the text library, which can be used to perform subsequent operations on the text library.

Examples

Sample requests

```
{
 "Name": "test1sdkdk",
 "BizTypes": [],
 "Category": "BLACK",
 "ServiceModule": "open_api",
 "ResourceType": "TEXT",
 "LibType": "textKeyword"
}
```

Sample success responses

```
{
 "RequestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200,
 "Id":123
}
```

6.3. Modify a text library

This topic describes how to call the UpdateKeywordLib operation to modify the basic information of a specified custom text library. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: UpdateKeywordLib

You can call this operation to modify the basic information of a specified text library.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the text library.
Name	String	Yes	The name of the text library.
BizTypes	List	No	The business type of the custom text library that you set on the Moderation Policy tab (by choosing **Settings** > Moderation API).
Enable	Boolean	No	Specifies whether to enable the text library. A value of true indicates that the text library is enabled, and a value of false indicates that the text library is disabled.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Id": 2147,
 "Name": "test2sdkdk",
 "BizTypes": []
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

6.4. Delete a text library

This topic describes how to call the DeleteKeywordLib operation to delete a specified custom text library. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: DeleteKeywordLib

You can call this operation to delete a specified text library.

ⓘ **Note** Deleted text libraries cannot be restored. Proceed with caution.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the text library to delete.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Id": 2147
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```

6.5. Add text

This topic describes how to call the CreateKeyword operation to add terms to a specified custom text library. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: CreateKeyword

You can call this operation to add terms to a specified text library.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
KeywordLibId	Numeric	Yes	The primary key ID of the text library.
Keywords	String	Yes	The terms to add.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Parameter	Type	Required	Description
SuccessCount	Numeric	Yes	The number of terms that have been added.
InvalidKeywordList	String	Yes	The invalid terms, that is, the terms that failed to be added.

Examples

Sample requests

```
{
 "KeywordLibId": "2147",
 "Keywords": [test]
}
```

Sample success responses

```
{
 "code": 200,
 "data": {
  "InvalidKeywordList": [
   "test"
  ],
  "SuccessCount": 0
 }
}
```

6.6. Query text

This topic describes how to call the DescribeKeyword operation to query terms in a specified custom text library in fuzzy mode. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: DescribeKeyword

You can call this operation to query terms by page in a specified text library.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description
KeywordLibId	Numeric	Yes	The primary key ID of the text library.
Keyword	String	No	The term to query. Fuzzy matching is supported. **Note** If you do not specify this parameter, all terms in the text library are returned.
TotalCount	Numeric	No	The number of terms to return.
CurrentPage	String	No	The number of the page to return.

Parameter	Type	Required	Description
PageSize	Numeric string	No	The number of entries to return on each page.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Parameter	Type	Required	Description
TotalCount	Numeric	No	The number of returned terms.
CurrentPage	String	No	The page number of the returned page.
PageSize	Numeric string	No	The number of entries returned per page.
KeywordList	JSON array	Yes	The terms. For more information about the parameter structure, see KeywordList.

KeywordList

Parameter	Type	Required	Description
Id	Numeric	Yes	The primary key ID of the term.
CreateTime	String	Yes	The time when the term was created.
Keyword	String	Yes	The term.
HitCount	Numeric	Yes	The number of times that the term was hit.

Examples

Sample requests

```
{
  "KeywordLibId": "2147",
}
```

Sample success responses

```
{
 "code": 200,
 "requestId": "@guid",
 "data": {
  "TotalCount": 1000,
  "PageSize": 20,
  "CurrentPage": 1,
  "KeywordList": [
   {
    "Keyword": "USA",
    "HitCount": 101,
    "CreateTime": "2018-04-19 20:12:30 +0800",
    "Id": 1
   },
   {
    "Keyword": "Donald Trump",
    "HitCount": 102,
    "CreateTime": "2018-04-19 20:12:30 +0800",
    "Id": 2
   }
  ]
 }
}
```

6.7. Delete text

This topic describes how to call the DeleteKeyword operation to delete terms from a specified custom text library. You can use text libraries for text anti-spam. The effect of managing custom text libraries by calling API operations is the same as that of managing custom text libraries in the Alibaba Cloud Content Moderation console.

Description

Operation: DeleteKeyword

You can call this operation to delete terms or text patterns from a specified text library.

Note Deleted terms cannot be restored. Proceed with caution.

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 following table describes the parameters in the request body.

Parameter	Type	Required	Description

Parameter	Type	Required	Description
Ids	List	No	The IDs of the terms to delete. **Note** You must specify either the Ids or Keywords parameter.
Keywords	List	No	The terms to delete. **Note** You must specify either the Ids or Keywords parameter.
KeywordLibId	String	Yes	The primary key ID of the text library.

Response parameters

For more information about common response parameters that this operation returns, see Common response parameters.

Examples

Sample requests

```
{
 "Id": [1,2]
 "KeywordLibId": 2147
}
```

Sample success responses

```
{
 "requestId": "795D8871-4889-4C0F-A8B1-C7D2B990FF61",
 "code": 200
}
```