

ALIBABA CLOUD

# Alibaba Cloud

内容安全

SDK User Manual

Document Version: 20200925

 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
 <b>Danger</b>	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 <b>Danger:</b> Resetting will result in the loss of user configuration data.
 <b>Warning</b>	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 <b>Warning:</b> Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 <b>Notice</b>	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 <b>Notice:</b> If the weight is set to 0, the server no longer receives new requests.
 <b>Note</b>	A note indicates supplemental instructions, best practices, tips, and other content.	 <b>Note:</b> You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click <b>Settings&gt; Network&gt; Set network type</b> .
<b>Bold</b>	<b>Bold</b> formatting is used for buttons, menus, page names, and other UI elements.	Click <b>OK</b> .
<b>Courier font</b>	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.SDK overview	06
2.SDK parameters	08
3.Java SDK	09
3.1. Installation	09
3.2. Initialization	10
3.3. Image moderation	12
3.4. Video moderation	30
3.5. Text anti-spam	47
3.6. OCR	52
3.7. Custom text library	56
3.8. Custom image library	62
4.Python SDK	68
4.1. Installation	68
4.2. Initialization	68
4.3. Image moderation	69
4.4. Video moderation	77
4.5. Text anti-spam	87
4.6. OCR	89
5.PHP SDK	92
5.1. Installation	92
5.2. Initialization	92
5.3. Image moderation	93
5.4. Video moderation	101
5.5. Text anti-spam	111
5.6. OCR	114
6.SDKs in other languages	117

---

7.Sample code download .....	118
8.SDK release notes .....	119


# 1.SDK overview

This topic describes how to use the software development kits (SDKs) that Alibaba Cloud Content Moderation provides in different development languages for you to call the Content Moderation API.

## SDK instructions

Before using the SDKs, read the [Content Moderation API documentation](#) to understand the feature of each Content Moderation API operation.

- In the SDKs, we expose the image moderation features, such as pornography detection, terrorist content detection, Optical Character Recognition (OCR), and logo detection, as one API. This API provides the following operations for you to call:
  - ImageSyncScanRequest operation: a synchronous operation to moderate images. You can call this operation to moderate multiple images at a time. The operation returns the moderation results of all images in real time. We recommend that you call this operation to moderate one image each time.
  - ImageAsyncScanRequest operation: an asynchronous operation to moderate images. You can call this operation to moderate multiple images at a time. If you submit multiple tasks to moderate multiple images, the operation returns a task ID for each image. You can use the task ID to poll the moderation result of each image. We recommend that you call this operation to moderate multiple images at a time.
- In the SDKs, we expose the video moderation features, such as pornography detection, terrorist content detection, and logo detection, as one API. This API provides the following operations for you to call:
  - VideoSyncScanRequest operation: a synchronous operation to moderate videos. To call this operation to moderate a video, you must convert the video to a sequence of image frames and submit the image frames for moderation. We recommend that you do not call this operation to moderate videos.
  - VideoAsyncScanRequest operation: an asynchronous operation to moderate videos. You can call this operation to submit either a video or a sequence of image frames captured from the video for moderation. After you submit a video moderation task, the operation returns a task ID for the task. You can use the task ID to poll the moderation result. Alternatively, you can configure a callback URL to receive a callback notification of the moderation result. We recommend that you call this operation to moderate videos.
- Audio anti-spam: The SDKs can detect spam in audio streams and audio files. Currently, the SDKs provide only an asynchronous operation to moderate audio content. After you submit an audio moderation task, the operation returns a task ID for the task. You can use the task ID to poll the moderation result. Alternatively, you can configure a callback URL to receive a callback notification of the moderation result.
- Text anti-spam: The SDKs provide only a synchronous operation to detect spam in text. You can send a request to moderate one or more pieces of text.

 **Note** If you specify multiple moderation scenarios when calling an operation, such as an image moderation operation, the expenses of all scenarios are calculated separately and summed up. The expense of a scenario equals the product of the unit price in that scenario and the quantity of scanned content.

## Development preparations


- Prepare the development environment for the Content Moderation SDK in the specific language that you choose.
- Content Moderation provides the following SDKs in three different languages:
  - [Java SDK](#)
  - [PHP SDK](#)
  - [Python SDK](#)
- Download the SDK sample code.

Click to download the [green-sdk-sample\\_doc](#) package.

The sample code provides complete call examples of Java, PHP, and Python SDKs for your reference.

- Obtain third-party SDKs in other languages for reference.

If you use a development language other than Java, PHP, or Python, we recommend that you directly send an HTTP request to call the corresponding Content Moderation API operation. We have also listed some content moderation SDKs compiled by third-party developers for your reference. The third-party SDKs are available in the following languages: C#, C++, Node.js, and Go. For more information, see [SDKs in other languages](#).

 **Note** The third-party SDKs are only for your reference. Alibaba Cloud does not provide follow-up maintenance for them.

## 2.SDK parameters

This topic describes the scenes, label, and suggestion parameters used in Content Moderation SDKs.

### Parameters scenes and label

In a content moderation request, you must set the scenes parameter to specify the moderation scenario. In the response, the label parameter indicates the risk category of the moderated object.

Content Moderation SDKs allow you to specify multiple moderation scenarios in a request. To view the values of the scenes parameter representing different moderation scenarios and the values of the label parameter in each moderation scenario, see the documentation of different Content Moderation API operations.

For example, if you want to call the ImageSyncScanRequest operation to moderate an image for pornography, specify *porn* in the scenes parameter. You can send requests to detect risky content in other moderation scenarios in a similar way.

You can also specify multiple moderation scenarios at the same time. For example, you can specify *porn* and *ad* in the scenes parameter if you want to moderate an image for pornography and ad violations.

### Parameter suggestion in the response

The suggestion parameter in the response indicates the recommended action on the moderated object if the server detected risky content in the moderated object.

- If the value of the scenes parameter is *porn*, *ad*, or *terrorism*, valid values of the suggestion parameter are as follows:
  - *pass*: The moderated object is normal.
  - *review*: The moderated object requires human review.
  - *block*: The moderated object contains violations and can be deleted or blocked.
- If the value of the scenes parameter is *qrcode*, valid values of the suggestion parameter are as follows:
  - *pass*: The moderated object does not contain a QR code.
  - *review*: The moderated object contains a QR code and requires human review. In this case, check the value of the qrcodeData parameter in the response to obtain the detected content.




# 3. Java SDK

## 3.1. Installation

This topic uses `aliyun-java-sdk-green` 3.6.1 as an example to describe how to install Content Moderation SDK for Java.

### Prerequisites

Java 1.6 or later is installed.

 **Note** You can run the `java -version` command to check the Java version.

### Install the SDK

To use `aliyun-java-sdk-green` in your Maven project, you only need to add relevant dependencies to the `pom.xml` file. For example, if you want to use `aliyun-java-sdk-green` 3.6.1, add the following content to `<dependencies>` :

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.1.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-green</artifactId>
  <version>3.6.1</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.51</version>
</dependency>
<dependency>
  <groupId>com.aliyun.oss</groupId>
  <artifactId>aliyun-sdk-oss</artifactId>
  <version>2.8.3</version>
</dependency>
```

Some of the Java sample code is compiled based on Java dependencies to read files and convert strings. You can add the following content to `<dependencies>` as needed:

```
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.4</version>
</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.10</version>
</dependency>
```

Download and import the [Extension.Uploader utility class](#) into your project if you submit a local or binary file for content moderation.

## 3.2. Initialization

IAcsClient is the Java client of aliyun-java-sdk-green, the SDK for Java. Before you use aliyun-java-sdk-green to make an API call, you must initialize an IAcsClient instance and modify the IClientProfile configurations as needed.

### Usage notes

When you create an IAcsClient instance, you must specify the region and the AccessKey ID and AccessKey secret of your Alibaba Cloud account.

- For more information about the supported regions, see [Endpoint](#).
- For more information about how to obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account, see [Create an AccessKey pair](#).

### Create an IAcsClient instance to moderate images, audios, videos, and text

The following regions are supported:

- China (Shanghai): cn-shanghai
- China (Beijing): cn-beijing
- Singapore (Singapore): ap-southeast-1
- US (Silicon Valley): us-west-1

 **Note** Only specific algorithmic models are available in the Singapore (Singapore) and US (Silicon Valley) regions.

Use the following code to create an IAcsClient instance to moderate images, audios, videos, and text:

```
/**
 * Set the ALIYUN_ACCESS_KEY_ID and ALIYUN_ACCESS_KEY_SECRET parameters to the AccessKey ID
 and AccessKey secret of your Alibaba Cloud account, respectively.
 * Valid values of the REGION_ID parameter: cn-shanghai, cn-beijing, ap-southeast-1, and us-west-1.
 Other regions are not supported.
 * Security risks may arise if you log on with the AccessKey pair of an Alibaba Cloud account becaus
 e the account has permissions on all API operations. We recommend that you log on as a Resource Acc
 ess Management (RAM) user to call API operations or perform routine operations and maintenance (O
 &M).
 */
String ALIYUN_ACCESS_KEY_ID = "<yourAccessKeyId>"
String ALIYUN_ACCESS_KEY_SECRET = "<yourAccessKeySecret>";
String REGION_ID = "cn-shanghai";

IClientProfile profile = DefaultProfile.getProfile(REGION_ID, ALIYUN_ACCESS_KEY_ID, ALIYUN_ACCESS
_KEY_SECRET);
IAcsClient recognitionClient = new DefaultAcsClient(profile)
```

## Create an IAcsClient instance to manage custom image libraries, term libraries, and text pattern libraries

Only the China (Shanghai) region is supported. The region ID is cn-shanghai. The server can automatically synchronize data if you initiate a request in a region other than the China (Shanghai) region.

Use the following code to create an IAcsClient instance to manage custom image libraries, term libraries, and text pattern libraries:

```

/**
 * Set the ALIYUN_ACCESS_KEY_ID and ALIYUN_ACCESS_KEY_SECRET parameters to the AccessKey ID
 and AccessKey secret of your Alibaba Cloud account, respectively.
 * Set the REGION_ID parameter to cn-shanghai. Other regions are not supported.
 * Security risks may arise if you log on with the AccessKey pair of an Alibaba Cloud account becaus
 e the account has permissions on all API operations. We recommend that you log on as a RAM user to c
 all API operations or perform routine O&M.
 */
String ALIYUN_ACCESS_KEY_ID = "<yourAccessKeyId>"
String ALIYUN_ACCESS_KEY_SECRET = "<yourAccessKeySecret>";
String REGION_ID = "cn-shanghai";

IClientProfile profile = DefaultProfile.getProfile(REGION_ID, ALIYUN_ACCESS_KEY_ID, ALIYUN_ACCESS
_KEY_SECRET);
IAcsClient managementClient = new DefaultAcsClient(profile)

```

## 3.3. Image moderation

This topic describes how to use Content Moderation SDK for Java to moderate images for risky content.

### Background information

Content Moderation SDK for Java supports both synchronous and asynchronous image moderation.

Moderation method	Moderation object	Method for obtaining moderation results
<p>Recommended: Submit synchronous image moderation tasks.</p> <p><b>ImageSyncScanRequest</b></p>	<p>You can submit the URL of an online image, the URL of a local image, or a binary image stream for image moderation.</p>	<p>You can obtain moderation results in real time.</p>

Moderation method	Moderation object	Method for obtaining moderation results
<p>Submit asynchronous image moderation tasks.</p> <p><b>ImageAsyncScanRequest</b></p>	<p>You can submit the URL of an online image, the URL of a local image, or a binary image stream for image moderation.</p>	<p>You can use one of the following methods to obtain moderation results:</p> <ul style="list-style-type: none"> <li>When you submit asynchronous image moderation tasks, specify the callback URL by setting the <code>callback</code> parameter. After the tasks are processed, Content Moderation sends the moderation results to the specified callback URL.</li> <li>After you submit asynchronous image moderation tasks, you can call the <b>ImageAsyncScanResultsRequest</b> operation to poll the moderation results.</li> </ul>

References:

- [Moderate images synchronously](#)
- [Moderate images asynchronously](#)

### Before you begin

Before you call operations, make the following preparations:

- Create an AccessKey pair for your Alibaba Cloud account. For more information, see [Create an AccessKey pair](#).
- Install Java dependencies. For more information, see [Installation](#).
- (Optional)Download and import the **Extension.Uploader utility class** into your project if you submit a local image or a binary image stream for image moderation.

### Recommended: Submit synchronous image moderation tasks

Operation	Description	Supported region
ImageSyncScanRequest	Sends synchronous requests to moderate images for risky content in multiple moderation scenarios, including pornography, terrorist content, ad, QR code, undesirable scene, and logo detection.	<ul style="list-style-type: none"> <li>• <b>cn-shanghai</b>: China (Shanghai)</li> <li>• <b>cn-beijing</b>: China (Beijing)</li> <li>• <b>cn-shenzhen</b>: China (Shenzhen)</li> <li>• <b>ap-southeast-1</b>: Singapore (Singapore)</li> <li>• <b>us-west-1</b>: US (Silicon Valley)</li> </ul>

Sample code

- Submit the URL of an online image for image moderation

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.green.model.v20180509.ImageSyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile
            .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile
            .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        ImageSyncScanRequest imageSyncScanRequest = new ImageSyncScanRequest();
        // Specify the response format of the operation.
        imageSyncScanRequest.setAcceptFormat(FormatType.JSON);
        // Specify the request method.
        imageSyncScanRequest.setMethod(MethodType.POST);
        imageSyncScanRequest.setEncoding("utf-8");
        // Both HTTP and HTTPS are supported.
        imageSyncScanRequest.setProtocol(ProtocolType.HTTP);

        JSONObject httpBody = new JSONObject();
        /**
         * Specify the moderation scenario. The system charges you based on the moderation scenario
         that you specify.
         * You can send a request to moderate multiple images at a time and specify multiple moderatio
```

n scenarios for each image. The expenses of all scenarios are separately calculated and summed up

\* For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.

\* In the scenes parameter, a value of porn indicates that the server detects pornography.

\*/

```
httpBody.put("scenes", Arrays.asList("porn"));
```

```
/**
```

\* Create a task for each image to be moderated.

\* If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.

\* Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.

\* In this example, a single image is to be moderated. If you want to moderate multiple images at a time, create a task for each image to be moderated.

\*/

```
JSONObject task = new JSONObject();
```

```
task.put("dataId", UUID.randomUUID().toString());
```

```
// Specify the image URL.
```

```
task.put("url", "http://xxx.test.jpg");
```

```
task.put("time", new Date());
```

```
httpBody.put("tasks", Arrays.asList(task));
```

```
imageSyncScanRequest.setHttpContent(org.apache.commons.codec.binary.StringUtils.getBytesUtf8(httpBody.toJSONString()),
```

```
"UTF-8", FormatType.JSON);
```

```
/**
```

\* Specify the connection timeout and read timeout. The timeout period for the server to complete an image moderation request is 10s.

\* If you set the read timeout to a period shorter than 10s, the server is prone to generate a read timeout error during request processing.

\*/

```
imageSyncScanRequest.setConnectTimeout(3000);
```

```
imageSyncScanRequest.setReadTimeout(10000);
```

```
HttpResponse httpResponse = null;
```

```
try {
```

```

        httpResponse = client.doAction(imageSyncScanRequest);
    } catch (Exception e) {
        e.printStackTrace();
    }

    // The results that are returned after the server receives and processes your request.
    if (httpResponse != null && httpResponse.isSuccess()) {
        JSONObject scrResponse = JSON.parseObject(org.apache.commons.codec.binary.StringUtils.newStringUtf8(httpResponse.getHttpContent()));
        System.out.println(JSON.toJSONString(scrResponse, true));
        int requestCode = scrResponse.getIntValue("code");
        // The moderation results of each image.
        JSONArray taskResults = scrResponse.getJSONArray("data");
        if (200 == requestCode) {
            for (Object taskResult : taskResults) {
                // The moderation results of a single image.
                int taskCode = ((JSONObject) taskResult).getIntValue("code");
                // The moderation result of the image in a moderation scenario. If you specified multiple
                moderation scenarios in the request, the moderation result of the image in each scenario is returned.
                JSONArray sceneResults = ((JSONObject) taskResult).getJSONArray("results");
                if (200 == taskCode) {
                    for (Object sceneResult : sceneResults) {
                        String scene = ((JSONObject) sceneResult).getString("scene");
                        String suggestion = ((JSONObject) sceneResult).getString("suggestion");
                        // Take a further action on the image based on the values of the scene and suggestion parameters.
                        //do something
                        System.out.println("scene = [" + scene + "]");
                        System.out.println("suggestion = [" + suggestion + "]");
                    }
                } else {
                    // A single image failed to be moderated. Analyze the failure based on the actual situation.
                    System.out.println("task process fail. task response:" + JSON.toJSONString(taskResult));
                }
            }
        } else {
            /**
             * Your whole request failed to be processed. Analyze the failure based on the actual situation.

```



```
tion.  
    */  
    System.out.println("the whole image scan request failed. response:" + JSON.toJSONString(s  
crResponse));  
    }  
    }  
    }  
}
```

- Submit the URL of a local image for image moderation

```
import com.alibaba.fastjson.JSON;  
import com.alibaba.fastjson.JSONArray;  
import com.alibaba.fastjson.JSONObject;  
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.green.extension.uploader.ClientUploader;  
import com.aliyuncs.green.model.v20180509.ImageSyncScanRequest;  
import com.aliyuncs.http.FormatType;  
import com.aliyuncs.http.HttpResponse;  
import com.aliyuncs.http.MethodType;  
import com.aliyuncs.http.ProtocolType;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.profile.IClientProfile;  
  
import java.util.*;  
  
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        IClientProfile profile = DefaultProfile  
            .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");  
        DefaultProfile  
            .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");  
        IAcsClient client = new DefaultAcsClient(profile);  
  
        ImageSyncScanRequest imageSyncScanRequest = new ImageSyncScanRequest();  
        // Specify the response format of the operation.  
        imageSyncScanRequest.setAcceptFormat(FormatType.JSON);  
        // Specify the request method.
```

```
imageSyncScanRequest.setMethod(MethodType.POST);
imageSyncScanRequest.setEncoding("utf-8");
// Both HTTP and HTTPS are supported.
imageSyncScanRequest.setProtocol(ProtocolType.HTTP);

JSONObject httpBody = new JSONObject();
/**
 * Specify the moderation scenario. The system charges you based on the moderation scenario
that you specify.
 * You can send a request to moderate multiple images at a time and specify multiple moderatio
n scenarios for each image. The expenses of all scenarios are separately calculated and summed up
.
 * For example, if you moderate two images for both pornography and terrorist content, you are
charged for moderating two images for pornography and two images for terrorist content.
 * In the scenes parameter, a value of porn indicates that the server detects pornography.
 */
httpBody.put("scenes", Arrays.asList("porn"));

/**
 * If you want to moderate a local image, use the following code snippet to generate a URL base
d on the storage path of the image.
 * Then, submit the generated URL of the image to the server.
 */
String url = null;
ClientUploader clientUploader = ClientUploader.getImageClientUploader(profile, false);
try{
    url = clientUploader.uploadFile("d:/test.jpg");
}catch (Exception e){
    e.printStackTrace();
}

/**
 * Create a task for each image to be moderated.
 * If you moderate multiple images in a request, the total response time that the server spends
processing the request starts from when the request is initiated to when the last image is moderat
ed.
 * Generally, the average response time of moderating multiple images in a request is longer th
an that of moderating a single image. The more images you submit at a time, the higher the probab
ility that the average response time will be extended.
 * In this example, a single image is to be moderated. If you want to moderate multiple images a
```

t a time, create a task for each image to be moderated.

```
*/
JSONObject task = new JSONObject();
task.put("dataId", UUID.randomUUID().toString());

// Set the url parameter to the image URL that is submitted to the server.
task.put("url", url);
task.put("time", new Date());
httpBody.put("tasks", Arrays.asList(task));

imageSyncScanRequest.setHttpContent(org.apache.commons.codec.binary.StringUtils.getBytes
Utf8(httpBody.toJSONString()),
    "UTF-8", FormatType.JSON);

/**
 * Specify the connection timeout and read timeout. The timeout period for the server to comple
te an image moderation request is 10s.
 * If you set the read timeout to a period shorter than 10s, the server is prone to generate a rea
d timeout error during request processing.
 */
imageSyncScanRequest.setConnectTimeout(3000);
imageSyncScanRequest.setReadTimeout(10000);
HttpResponse httpResponse = null;
try {
    httpResponse = client.doAction(imageSyncScanRequest);
} catch (Exception e) {
    e.printStackTrace();
}

// The results that are returned after the server receives and processes your request.
if (httpResponse != null && httpResponse.isSuccess()) {
    JSONObject scrResponse = JSON.parseObject(org.apache.commons.codec.binary.StringUtils.ne
wStringUtf8(httpResponse.getHttpContent()));
    System.out.println(JSON.toJSONString(scrResponse, true));
    int requestCode = scrResponse.getIntValue("code");
    // The moderation results of each image.
    JSONArray taskResults = scrResponse.getJSONArray("data");
    if (200 == requestCode) {
        for (Object taskResult : taskResults) {
            // The moderation results of a single image.
            int taskCode = ((JSONObject) taskResult).getIntValue("code");
```

```

int taskCode = ((JSONObject) taskResult).getIntValue("code");
// The moderation result of the image in a moderation scenario. If you specified multiple
moderation scenarios in the request, the moderation result of the image in each scenario is returned.

JSONArray sceneResults = ((JSONObject) taskResult).getJSONArray("results");
if (200 == taskCode) {
    for (Object sceneResult : sceneResults) {
        String scene = ((JSONObject) sceneResult).getString("scene");
        String suggestion = ((JSONObject) sceneResult).getString("suggestion");
        // Take a further action on the image based on the values of the scene and suggestion parameters.
        //do something
        System.out.println("scene = [" + scene + "];");
        System.out.println("suggestion = [" + suggestion + "];");
    }
} else {
    // A single image failed to be moderated. Analyze the failure based on the actual situation.
    System.out.println("task process fail. task response:" + JSON.toJSONString(taskResult));
}
}
} else {
    /**
     * Your whole request failed to be processed. Analyze the failure based on the actual situation.
     */
    System.out.println("the whole image scan request failed. response:" + JSON.toJSONString(scrResponse));
}
}
}
}

```

- Submit a binary image stream for image moderation

```

import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;

```

```
import com.aliyuncs.green.extension.uploader.ClientUploader;
import com.aliyuncs.green.model.v20180509.ImageSyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import org.apache.commons.io.FileUtils;

import java.io.File;
import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile
            .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile
            .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        ImageSyncScanRequest imageSyncScanRequest = new ImageSyncScanRequest();
        // Specify the response format of the operation.
        imageSyncScanRequest.setAcceptFormat(FormatType.JSON);
        // Specify the request method.
        imageSyncScanRequest.setMethod(MethodType.POST);
        imageSyncScanRequest.setEncoding("utf-8");
        // Both HTTP and HTTPS are supported.
        imageSyncScanRequest.setProtocol(ProtocolType.HTTP);

        JSONObject httpBody = new JSONObject();
        /**
         * Specify the moderation scenario. The system charges you based on the moderation scenario
         that you specify.
         * You can send a request to moderate multiple images at a time and specify multiple moderatio
         n scenarios for each image. The expenses of all scenarios are separately calculated and summed up
         .
         * For example, if you moderate two images for both pornography and terrorist content, you are
         charged for moderating two images for pornography and two images for terrorist content.
        */
    }
}
```

```
* In the scenes parameter, a value of porn indicates that the server detects pornography.
*/
httpBody.put("scenes", Arrays.asList("porn"));

/**
 * If you want to moderate a local image, use the following code snippet to generate a URL based on the storage path of the image.
 * Then, submit the generated URL of the image to the server.
 */
ClientUploader clientUploader = ClientUploader.getImageClientUploader(profile, false);
byte[] imageBytes = null;
String url = null;
try{
    // Read and convert a local image to binary data and submit the binary data for moderation. In the actual code, directly use the binary data of your image.
    imageBytes = FileUtils.readFileToByteArray(new File("/Users/01fb4ab6420b5f34623e13b82b51ef87.jpg"));
    // Upload the binary stream to the server.
    url = clientUploader.uploadBytes(imageBytes);
}catch (Exception e){
    e.printStackTrace();
}

/**
 * Create a task for each image to be moderated.
 * If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
 * Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
 * In this example, a single image is to be moderated. If you want to moderate multiple images at a time, create a task for each image to be moderated.
 */
JSONObject task = new JSONObject();
task.put("dataId", UUID.randomUUID().toString());

// Set the url parameter to the image URL that is submitted to the server.
task.put("url", url);
task.put("time", new Date());
httpBody.put("tasks", Arrays.asList(task));
```

```
httpBody.put("tasks", Arrays.asList(task));

imageSyncScanRequest.setHttpContent(org.apache.commons.codec.binary.StringUtils.getBytes
Utf8(httpBody.toJSONString()),
    "UTF-8", FormatType.JSON);

/**
 * Specify the connection timeout and read timeout. The timeout period for the server to comple
te an image moderation request is 10s.
 * If you set the read timeout to a period shorter than 10s, the server is prone to generate a rea
d timeout error during request processing.
 */
imageSyncScanRequest.setConnectTimeout(3000);
imageSyncScanRequest.setReadTimeout(10000);
HttpResponse httpResponse = null;
try {
    httpResponse = client.doAction(imageSyncScanRequest);
} catch (Exception e) {
    e.printStackTrace();
}

// The results that are returned after the server receives and processes your request.
if (httpResponse != null && httpResponse.isSuccess()) {
    JSONObject scrResponse = JSON.parseObject(org.apache.commons.codec.binary.StringUtils.ne
wStringUtf8(httpResponse.getHttpContent()));
    System.out.println(JSON.toJSONString(scrResponse, true));
    int requestCode = scrResponse.getIntValue("code");
    // The moderation results of each image.
    JSONArray taskResults = scrResponse.getJSONArray("data");
    if (200 == requestCode) {
        for (Object taskResult : taskResults) {
            // The moderation results of a single image.
            int taskCode = ((JSONObject) taskResult).getIntValue("code");
            // The moderation result of the image in a moderation scenario. If you specified multiple
moderation scenarios in the request, the moderation result of the image in each scenario is returne
d.
            JSONArray sceneResults = ((JSONObject) taskResult).getJSONArray("results");
            if (200 == taskCode) {
                for (Object sceneResult : sceneResults) {
                    String scene = ((JSONObject) sceneResult).getString("scene");
                    String suggestion = ((JSONObject) sceneResult).getString("suggestion");
```

```
        // Take a further action on the image based on the values of the scene and suggest
ion parameters.
        //do something
        System.out.println("scene = [" + scene + "]");
        System.out.println("suggestion = [" + suggestion + "]");
    }
} else {
    // A single image failed to be moderated. Analyze the failure based on the actual situa
tion.
    System.out.println("task process fail. task response:" + JSON.toJSONString(taskResult
));
}
}
} else {
    /**
     * Your whole request failed to be processed. Analyze the failure based on the actual situa
tion.
     */
    System.out.println("the whole image scan request failed. response:" + JSON.toJSONString(s
crResponse));
}
}
}
}
```


## Submit asynchronous image moderation tasks

This section describes how to use Content Moderation SDK for Java to call the `ImageAsyncScanRequest` operation and moderate images for risky content. You can send an asynchronous request to submit image moderation tasks. When you submit the request, you can specify the callback URL to receive moderation results by setting the callback parameter. Alternatively, you can call the `ImageAsyncScanResultsRequest` operation to poll the moderation results after Content Moderation processes the image moderation tasks.

Same as synchronous image moderation, you can submit the URL of an online image, the URL of a local image, or a binary image stream for asynchronous image moderation. In this example, the URL of an online image is used.

Operation	Description	Supported region
-----------	-------------	------------------



Operation	Description	Supported region
ImageAsyncScanRequest	<p>Sends asynchronous requests to moderate images for risky content in multiple moderation scenarios, including pornography, terrorist content, ad, QR code, undesirable scene, and logo detection.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> <b>Note</b> You can call the <b>ImageAsyncScanResultsRequest</b> operation to poll the moderation results.</p> </div>	<ul style="list-style-type: none"> <li>• cn-shanghai: China (Shanghai)</li> <li>• cn-beijing: China (Beijing)</li> <li>• cn-shenzhen: China (Shenzhen)</li> <li>• ap-southeast-1: Singapore (Singapore)</li> <li>• us-west-1: US (Silicon Valley)</li> <li>•</li> </ul>

### Sample code

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.green.model.v20180509.ImageAsyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.Arrays;
import java.util.Date;
import java.util.UUID;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your Access Key secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        ImageAsyncScanRequest imageAsyncScanRequest = new ImageAsyncScanRequest();
```

```
imageAsyncScanRequest.setAcceptFormat(FormatType.JSON);
// Specify the request method.
imageAsyncScanRequest.setMethod(MethodType.POST);
imageAsyncScanRequest.setEncoding("utf-8");
// Both HTTP and HTTPS are supported.
imageAsyncScanRequest.setProtocol(ProtocolType.HTTP);

JSONObject httpBody = new JSONObject();
/**
 * Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.
 * You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are separately calculated and summed up.
 * For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
 * In the scenes parameter, a value of porn indicates that the server detects pornography.
 */
httpBody.put("scenes", Arrays.asList("porn"));
httpBody.put("callback", "http://xxx.xxx.xx/xxx.json");
httpBody.put("seed", "yourPersonalSeed");

/**
 * Create a task for each image to be moderated. You can moderate a maximum of 50 images at a time. In this case, you must create 50 tasks.
 * If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
 * Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
 * In this example, a single image is to be moderated. If you want to moderate multiple images at a time, create a task for each image to be moderated.
 */
JSONObject task = new JSONObject();
task.put("dataId", UUID.randomUUID().toString());

// Specify the image URL.
task.put("url", "http://xxx.test.jpg");
task.put("time", new Date());
```

```
httpBody.put("tasks", Arrays.asList(task));

imageAsyncScanRequest.setHttpContent(org.apache.commons.codec.binary.StringUtils.getBytesU
tf8(httpBody.toJSONString()),
    "UTF-8", FormatType.JSON);

/**
 * Specify the connection timeout and read timeout. The timeout period for the server to complete
 an image moderation request is 10s.
 * If you set the read timeout to a period shorter than 10s, the server is prone to generate a read
 timeout error during request processing.
 */
imageAsyncScanRequest.setConnectTimeout(3000);
imageAsyncScanRequest.setReadTimeout(10000);
HttpResponse httpResponse = null;
try {
    httpResponse = client.doAction(imageAsyncScanRequest);
} catch (Exception e) {
    e.printStackTrace();
}

// The results that are returned after the server receives and processes your request.
if (httpResponse != null && httpResponse.isSuccess()) {
    JSONObject scrResponse = JSON.parseObject(org.apache.commons.codec.binary.StringUtils.new
StringUtf8(httpResponse.getHttpContent()));
    System.out.println(JSON.toJSONString(scrResponse, true));
    int requestCode = scrResponse.getIntValue("code");
    // The moderation results of each image.
    JSONArray taskResults = scrResponse.getJSONArray("data");
    if (200 == requestCode) {
        for (Object taskResult : taskResults) {
            // The moderation results of a single image.
            int taskCode = ((JSONObject) taskResult).getIntValue("code");
            // The moderation result of the image in a moderation scenario. If you specified multiple m
oderation scenarios in the request, the moderation result of the image in each scenario is returned.
            JSONArray sceneResults = ((JSONObject) taskResult).getJSONArray("results");
            if (200 == taskCode) {
                // Save the task ID, which is used to poll the moderation results.
                System.out.println(((JSONObject)taskResult).getString("taskId"));
            } else {
                // A single image failed to be moderated. Analyze the failure based on the actual situati
```



```
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your Access
Key secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.co
m");
        IAcsClient client = new DefaultAcsClient(profile);

        ImageAsyncScanResultsRequest imageAsyncScanResultsRequest = new ImageAsyncScanResultsR
equest();
        // Specify the response format of the operation.
        imageAsyncScanResultsRequest.setAcceptFormat(FormatType.JSON);
        // Specify the request method.
        imageAsyncScanResultsRequest.setMethod(MethodType.POST);
        imageAsyncScanResultsRequest.setEncoding("utf-8");
        // Both HTTP and HTTPS are supported.
        imageAsyncScanResultsRequest.setProtocol(ProtocolType.HTTP);

        List<String> taskIds = new ArrayList<String>();
        taskIds.add("img4hDosCHcrFk5jAMR80XWJN-1pZ@0p");
        imageAsyncScanResultsRequest.setHttpContent(JSON.toJSONString(taskIds).getBytes("UTF-8"), "
UTF-8", FormatType.JSON);

        /**
         * Specify the connection timeout and read timeout.
         */
        imageAsyncScanResultsRequest.setConnectTimeout(3000);
        imageAsyncScanResultsRequest.setReadTimeout(6000);

        try {
            HttpResponse httpResponse = client.doAction(imageAsyncScanResultsRequest);
```

```
if(httpResponse.isSuccess()){
    JSONObject scrResponse = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
    System.out.println(JSON.toJSONString(scrResponse, true));
    if (200 == scrResponse.getInteger("code")) {
        JSONArray taskResults = scrResponse.getJSONArray("data");
        for (Object taskResult : taskResults) {
            if(200 == ((JSONObject)taskResult).getInteger("code")){
                JSONArray sceneResults = ((JSONObject)taskResult).getJSONArray("results");
                for (Object sceneResult : sceneResults) {
                    String scene = ((JSONObject)sceneResult).getString("scene");
                    String suggestion = ((JSONObject)sceneResult).getString("suggestion");
                    // Take a further action on the image based on the values of the scene and suggestion parameters.
                    //do something
                }
            }else{
                System.out.println("task process fail:" + ((JSONObject)taskResult).getInteger("code"));
            }
        }
    } else {
        System.out.println("detect not success. code:" + scrResponse.getInteger("code"));
    }
} else{
    System.out.println("response not success. status:" + httpResponse.getStatus());
}
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
} catch (Exception e){
    e.printStackTrace();
}
}
```

## 3.4. Video moderation

This topic describes how to use Content Moderation SDK for Java to moderate videos for risky content. You can moderate both images and audios in videos.

## Background information

Content Moderation SDK for Java supports both synchronous and asynchronous video moderation.

Moderation method	Moderation object	Method for obtaining moderation results
Submit synchronous video moderation tasks. <b>VideoSyncScanRequest</b>	You can submit only a sequence of frames that are captured from a video for video moderation.	You can obtain moderation results in real time.
Recommended: Submit asynchronous video moderation tasks. <b>VideoAsyncScanRequest</b>	You can submit the URL of an online video, the URL of a local video, a binary video stream, a video live stream, or a sequence of frames that are captured from a video for video moderation.	You can use one of the following methods to obtain moderation results: <ul style="list-style-type: none"> <li>When you submit asynchronous video moderation tasks, specify the callback URL by setting the <code>callback</code> parameter. After the tasks are processed, Content Moderation sends the moderation results to the specified callback URL.</li> <li>After you submit asynchronous video moderation tasks, you can call the <b>VideoAsyncScanResultsRequest</b> operation to poll the moderation results.</li> </ul>

## Before you begin

Before you call operations, make the following preparations:

- Create an AccessKey pair for your Alibaba Cloud account. For more information, see [Create an AccessKey pair](#).
- Install Java dependencies. For more information, see [Installation](#).
- (Optional)Download and import the **Extension.Uploader utility class** into your project if you submit a local image or a binary image stream for image moderation.

## Recommended: Submit asynchronous video moderation tasks

Operation	Description	Supported region
-----------	-------------	------------------

Operation	Description	Supported region
VideoAsyncScanRequest	Sends asynchronous requests to moderate videos for risky content in multiple moderation scenarios, including pornography, terrorist content, ad, undesirable scene, and logo detection.	<ul style="list-style-type: none"> <li>• cn-shanghai: China (Shanghai)</li> <li>• cn-beijing: China (Beijing)</li> <li>• cn-shenzhen: China (Shenzhen)</li> <li>• ap-southeast-1: Singapore (Singapore)</li> <li>• us-west-1: US (Silicon Valley)</li> </ul>

### Sample code

- Submit the URL of an online video for video moderation

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.VideoAsyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        VideoAsyncScanRequest videoAsyncScanRequest = new VideoAsyncScanRequest();
        videoAsyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the operation.
        videoAsyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.
    }
}
```



```

List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
Map<String, Object> task = new LinkedHashMap<String, Object>();
task.put("dataId", UUID.randomUUID().toString());
task.put("url", "Enter the HTTP or HTTPS URL of the online video that is accessible from the Internet.");

tasks.add(task);
/**
 * Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.
 * By default, a frame is captured from the video per second. You can specify the frequency at which sequential frames are captured. The system charges you based on the number of frames that are captured from the video and the moderation scenarios for each frame.
 * For example, if 60 frames are captured from a 1-minute-long video and you want to moderate the video for both pornography and terrorist content, you are charged for moderating 60 frames for pornography and 60 frames for terrorist content.
 */
JSONObject data = new JSONObject();
data.put("scenes", Arrays.asList("porn", "terrorism"));
data.put("tasks", tasks);
data.put("callback", "http://xxx.xxx.xx/xxx.json");
data.put("seed", "yourPersonalSeed");

videoAsyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", FormatType.JSON);

/**
 * Specify the connection timeout and read timeout.
 */
videoAsyncScanRequest.setConnectTimeout(3000);
videoAsyncScanRequest.setReadTimeout(6000);
try {
    HttpResponse httpResponse = client.doAction(videoAsyncScanRequest);

    if(httpResponse.isSuccess()){
        JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(jsonObject, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
}

```

```
    }  
  } catch (ServerException e) {  
    e.printStackTrace();  
  } catch (ClientException e) {  
    e.printStackTrace();  
  }  
}  
}
```

- Submit the URL of a local video for video moderation

```
import com.alibaba.fastjson.JSON;  
import com.alibaba.fastjson.JSONObject;  
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.green.extension.uploader.ClientUploader;  
import com.aliyuncs.green.model.v20180509.VideoAsyncScanRequest;  
import com.aliyuncs.http.FormatType;  
import com.aliyuncs.http.HttpResponse;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.profile.IClientProfile;  
  
import java.util.*;  
  
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");  
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");  
        IAcsClient client = new DefaultAcsClient(profile);  
  
        VideoAsyncScanRequest videoAsyncScanRequest = new VideoAsyncScanRequest();  
        videoAsyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the operation.  
        videoAsyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.  
  
        /**
```

\* If you want to moderate a local video, use the following code snippet to generate a URL based on the storage path of the video. Then, submit the generated URL of the video to the server.

```
*/
String url = null;
ClientUploader uploader = ClientUploader.getVideoClientUploader(profile, false);
try{
    url = uploader.uploadFile("The absolute path of the local video");
}catch (Exception e){
    e.printStackTrace();
}
```

```
List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
Map<String, Object> task = new LinkedHashMap<String, Object>();
task.put("dataId", UUID.randomUUID().toString());
task.put("url", url);
```

```
tasks.add(task);
```

```
/**
```

\* Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.

\* By default, a frame is captured from the video per second. You can specify the frequency at which sequential frames are captured. The system charges you based on the number of frames that are captured from the video and the moderation scenarios for each frame.

\* For example, if 60 frames are captured from a 1-minute-long video and you want to moderate the video for both pornography and terrorist content, you are charged for moderating 60 frames for pornography and 60 frames for terrorist content.

```
*/
JSONObject data = new JSONObject();
data.put("scenes", Arrays.asList("porn", "terrorism"));
data.put("tasks", tasks);
data.put("callback", "http://xxx.xxx.xx/xxx.json");
data.put("seed", "yourPersonalSeed");
```

```
videoAsyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", ContentType.JSON);
```

```
/**
```

\* Specify the connection timeout and read timeout.

```
*/
```

```
videoAsyncScanRequest.setConnectTimeout(3000);
```

```
videoAsyncScanRequest.setReadTimeout(10000);
try {
    HttpResponse httpResponse = client.doAction(videoAsyncScanRequest);

    if(httpResponse.isSuccess()){
        JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(jsonObject, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

- Submit a binary video stream for video moderation

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.extension.uploader.ClientUploader;
import com.aliyuncs.green.model.v20180509.VideoAsyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs
```

```
.com");
IAcsClient client = new DefaultAcsClient(profile);

VideoAsyncScanRequest videoAsyncScanRequest = new VideoAsyncScanRequest();
videoAsyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of
the operation.
videoAsyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request
method.

/**
 * If you want to moderate a local video, use the following code snippet to generate a URL based
on the storage path of the video. Then, submit the generated URL of the video to the server.
 */
ClientUploader uploader = ClientUploader.getVideoClientUploader(profile, false);
byte[] videoBytes = null;
String url = null;
try{
    // Read and convert a local video to binary data and submit the binary data for moderation. In
the actual code, directly use the binary data of your video.
    videoBytes = FileUtils.readFileToByteArray(new File("/Users/01fb4ab6420b5f34623e13b82b51
ef87.mp4"));
    // Upload the binary stream to the server.
    url = uploader.uploadBytes(videoBytes);
}catch (Exception e){
    System.out.println("upload file to server fail.", e);
}

List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
Map<String, Object> task = new LinkedHashMap<String, Object>();
task.put("dataId", UUID.randomUUID().toString());
task.put("url", url);

tasks.add(task);
/**
 * Specify the moderation scenario. The system charges you based on the moderation scenario
that you specify.
 * By default, a frame is captured from the video per second. You can specify the frequency at
which sequential frames are captured. The system charges you based on the number of frames that
are captured from the video and the moderation scenarios for each frame.
 * For example, if 60 frames are captured from a 1-minute-long video and you want to moderate
the video for both pornography and terrorist content, you are charged for moderating 60 frames for
```

```
pornography and 60 frames for terrorist content.
    */
    JSONObject data = new JSONObject();
    data.put("scenes", Arrays.asList("porn", "terrorism"));
    data.put("tasks", tasks);
    data.put("callback", "http://xxx.xxx.xx/xxx.json");
    data.put("seed", "yourPersonalSeed");

    videoAsyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8",
    ContentType.JSON);

    /**
     * Specify the connection timeout and read timeout.
     */
    videoAsyncScanRequest.setConnectTimeout(3000);
    videoAsyncScanRequest.setReadTimeout(10000);
    try {
        HttpResponse httpResponse = client.doAction(videoAsyncScanRequest);

        if(httpResponse.isSuccess()){
            JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UT
            F-8"));
            System.out.println(JSON.toJSONString(jsonObject, true));
        }else{
            System.out.println("response not success. status:" + httpResponse.getStatus());
        }
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
}
```

- Submit a video live stream for video moderation

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
```

```
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.extension.uploader.ClientUploader;
import com.aliyuncs.green.model.v20180509.VideoAsyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        VideoAsyncScanRequest videoAsyncScanRequest = new VideoAsyncScanRequest();
        videoAsyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the operation.
        videoAsyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.

        List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
        Map<String, Object> task = new LinkedHashMap<String, Object>();
        task.put("dataId", UUID.randomUUID().toString());
        // Set the url parameter to the URL of your live stream.
        task.put("url", "http://xxxx/test.mp4");

        tasks.add(task);
    }
    /**
     * Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.
     * By default, a frame is captured from the video per second. You can specify the frequency at which sequential frames are captured. The system charges you based on the number of frames that are captured from the video and the moderation scenarios for each frame.
     * For example, if 60 frames are captured from a 1-minute-long video and you want to moderate the video for both pornography and terrorist content, you are charged for moderating 60 frames for pornography and 60 frames for terrorist content.
    */
}
```

```
*/
JSONObject data = new JSONObject();
data.put("scenes", Arrays.asList("porn", "terrorism"));
data.put("live", true);
data.put("tasks", tasks);
data.put("callback", "http://xxx.xxx.xx/xxx.json");
data.put("seed", "yourPersonalSeed");

videoAsyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8",
FormatType.JSON);

/**
 * Specify the connection timeout and read timeout.
 */
videoAsyncScanRequest.setConnectTimeout(3000);
videoAsyncScanRequest.setReadTimeout(10000);
try {
    HttpResponse httpResponse = client.doAction(videoAsyncScanRequest);

    if(httpResponse.isSuccess()){
        JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(jsonObject, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

- Submit a video live stream to moderate both the video images and audio

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
```



```
import com.aliyuncs.green.extension.uploader.ClientUploader;
import com.aliyuncs.green.model.v20180509.VideoAsyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        VideoAsyncScanRequest videoAsyncScanRequest = new VideoAsyncScanRequest();
        videoAsyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the operation.
        videoAsyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.

        List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
        Map<String, Object> task = new LinkedHashMap<String, Object>();
        task.put("dataId", UUID.randomUUID().toString());
        // Set the url parameter to the URL of your live stream.
        task.put("url", "http://xxx/test.mp4");

        tasks.add(task);
        /**
         * Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.
         * By default, a frame is captured from the video per second. You can specify the frequency at which sequential frames are captured. The system charges you based on the number of frames that are captured from the video and the moderation scenarios for each frame.
         * For example, if 60 frames are captured from a 1-minute-long video and you want to moderate the video for both pornography and terrorist content, you are charged for moderating 60 frames for pornography and 60 frames for terrorist content.
         */
    }
}
```


```
*/
JSONObject data = new JSONObject();
data.put("scenes", Arrays.asList("porn", "terrorism"));
data.put("live", true);
data.put("tasks", tasks);
data.put("callback", "http://xxx.xxx.xx/xxx.json");
data.put("seed", "yourPersonalSeed");
/**
 * In addition to moderating video images, if you want to moderate the audio in the video for ris
ky content, set the audioScenes parameter to antispam.
 * The expense of audio moderation equals the product of the video duration and the unit price
of audio anti-spam.
 */
data.put("audioScenes", Arrays.asList("antispam"));

videoAsyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", Forma
tType.JSON);

/**
 * Specify the connection timeout and read timeout.
 */
videoAsyncScanRequest.setConnectTimeout(3000);
videoAsyncScanRequest.setReadTimeout(10000);
try {
    HttpResponse httpResponse = client.doAction(videoAsyncScanRequest);

    if(httpResponse.isSuccess()){
        JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UT
F-8"));
        System.out.println(JSON.toJSONString(jsonObject, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

## Obtain asynchronous video moderation results

Operation	Description	Supported region
VideoAsyncScanResultsRequest	<p>Queries asynchronous video moderation results.</p> <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc;"> <p> <b>Note</b> Instead of calling this operation to poll the moderation results, we recommend that you set the callback parameter when you submit asynchronous video moderation tasks to receive the moderation results.</p> </div>	<ul style="list-style-type: none"> <li>• <b>cn-shanghai:</b> China (Shanghai)</li> <li>• <b>cn-beijing:</b> China (Beijing)</li> <li>• <b>cn-shenzhen:</b> China (Shenzhen)</li> <li>• <b>ap-southeast-1:</b> Singapore (Singapore)</li> <li>• <b>us-west-1:</b> US (Silicon Valley)</li> <li>•</li> </ul>

### Sample code

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.VideoAsyncScanResultsRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your Access Key secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);

        VideoAsyncScanResultsRequest videoAsyncScanResultsRequest = new VideoAsyncScanResultsRequest();
    }
}
```

```

quest(),
    videoAsyncScanResultsRequest.setAcceptFormat(FormatType.JSON);

    List<String> taskList = new ArrayList<String>();
    // Specify the task ID to query the results of an asynchronous video moderation task that you submit by calling the VideoAsyncScanRequest operation. The task ID is returned by the VideoAsyncScanRequest operation after you submit the task.
    taskList.add("vi3pnWxOlikyx6KJNtY7Naza-1pZ$MN");


    videoAsyncScanResultsRequest.setHttpContent(JSON.toJSONString(taskList).getBytes("UTF-8"), "UTF-8", FormatType.JSON);

    /**
     * Specify the connection timeout and read timeout.
     */
    videoAsyncScanResultsRequest.setConnectTimeout(3000);
    videoAsyncScanResultsRequest.setReadTimeout(6000);
    try {
        HttpResponse httpResponse = client.doAction(videoAsyncScanResultsRequest);
        if(httpResponse.isSuccess()){
            JSONObject jsonObject = JSON
                .parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
            System.out.println(JSON.toJSONString(jsonObject, true));
        }else{
            System.out.println("response not success. status:" + httpResponse.getStatus());
        }
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
}
}


```

## Submit synchronous video moderation tasks

Operation	Description	Supported region
-----------	-------------	------------------

Operation	Description	Supported region
VideoSyncScanRequest	<p>Sends synchronous requests to moderate videos for risky content.</p> <p> <b>Note</b> You can submit only a sequence of frames that are captured from a video for video moderation. To submit other types of videos, we recommend that you use the VideoAsyncScanRequest operation.</p>	<ul style="list-style-type: none"> <li>• <b>cn-shanghai</b>: China (Shanghai)</li> <li>• <b>cn-beijing</b>: China (Beijing)</li> <li>• <b>cn-shenzhen</b>: China (Shenzhen)</li> <li>• <b>ap-southeast-1</b>: Singapore (Singapore)</li> <li>• <b>us-west-1</b>: US (Silicon Valley)</li> <li>• </li> </ul>

### Sample code

 **Note** In this example, a sequence of frames that are captured from a video is to be moderated.

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.VideoSyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main3 {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "Your AccessKey ID", "Your Access Key secret");
        DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);
```

```
VideoSyncScanRequest videoSyncScanRequest = new VideoSyncScanRequest();
videoSyncScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the operation.
videoSyncScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.

List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
Map<String, Object> task = new LinkedHashMap<String, Object>();
task.put("dataId", UUID.randomUUID().toString());

List<Map<String, Object>> frames = new ArrayList<Map<String, Object>>();
Map<String, Object> frame1 = new LinkedHashMap<String, Object>();
frame1.put("offset", 0);
frame1.put("url", "https://img.alicdn.com/tfs/TB1k_g9L26H8KjJSspmXXb2WXXa-600-600.jpg");

Map<String, Object> frame2 = new LinkedHashMap<String, Object>();
frame2.put("offset", 5);
frame2.put("url", "http://pic12.nipic.com/20110221/6727421_210944911000_2.jpg");

Map<String, Object> frame3 = new LinkedHashMap<String, Object>();
frame3.put("offset", 10);
frame3.put("url", "http://rifleman-share.oss-cn-hangzhou.aliyuncs.com/test/%E6%AD%A3%E5%B8%B8/68d5883924c9e8cc88806a73bd7a8995.jpg");
frames.addAll(Arrays.asList(frame1, frame2, frame3));

task.put("frames", frames);
tasks.add(task);
/**
 * Specify the moderation scenario. The system charges you based on the moderation scenario that you specify.
 * By default, a frame is captured from the video per second. You can specify the frequency at which sequential frames are captured. The system charges you based on the number of frames that are captured from the video and the moderation scenarios for each frame.
 * For example, if 60 frames are captured from a 1-minute-long video and you want to moderate the video for both pornography and terrorist content, you are charged for moderating 60 frames for pornography and 60 frames for terrorist content.
 */
JSONObject data = new JSONObject();
data.put("scenes", Arrays.asList("porn", "terrorism"));
data.put("tasks", tasks);
```


```
videoSyncScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", FormatType.JSON);

/**
 * Specify the connection timeout and read timeout.
 */
videoSyncScanRequest.setConnectTimeout(3000);
videoSyncScanRequest.setReadTimeout(10000);
try {
    HttpResponse httpResponse = client.doAction(videoSyncScanRequest);

    if(httpResponse.isSuccess()){
        JSONObject jsonObject = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(jsonObject, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

## 3.5. Text anti-spam

This topic describes how to use the Java SDK to moderate text for spam such as pornography and terrorist content.

 **Note** Currently, the Java SDK only supports synchronous text moderation. You can send a request to moderate one or more pieces of text. The system charges you based on the number of pieces of text moderated. For more information, see [Billing methods](#).

### Before you begin

Before you call operations, make the following preparations:

- Create an AccessKey pair for your Alibaba Cloud account. For more information, see [Create an AccessKey pair](#).

- Install Java dependencies. For more information, see [Installation](#).
- (Optional) Download and import the [Extension.Uploader utility class](#) into your project if you submit a local image or a binary image stream for image moderation.

## Moderate text for spam

Text anti-spam allows you to add custom terms, such as brand terms of competitors. If the text being moderated contains the terms you add, the value of the suggestion parameter returned by the text anti-spam algorithm is block.

You can log on to the [Alibaba Cloud Content Moderation console](#) or call the corresponding operation to add terms. The terms are UTF-8-encoded. For more information about the text anti-spam operation, see [Moderate text for spam](#).

Operation	Feature	Region	Description
TextScanRequest	Scans text for spam.	<ul style="list-style-type: none"> <li>• <i>cn-shanghai: China (Shanghai)</i></li> <li>• <i>cn-beijing: China (Beijing)</i></li> <li>• <i>cn-shenzhen: China (Shenzhen)</i></li> <li>• <i>ap-southeast-1: Singapore</i></li> <li>• <i>us-west-1: US (Silicon Valley)</i></li> </ul>	This operation scans text for spam after you specify antisipam in the scenes parameter.

### Sample code

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyun.oss.ClientException;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.TextScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
```



```

IClientProfile profile = DefaultProfile
    .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile
    .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
IAcsClient client = new DefaultAcsClient(profile);
TextScanRequest textScanRequest = new TextScanRequest();
textScanRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the oper
ation.
textScanRequest.setHttpContentType(FormatType.JSON);
textScanRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method
.
textScanRequest.setEncoding("UTF-8");
textScanRequest.setRegionId("cn-shanghai");
List<Map<String, Object>> tasks = new ArrayList<Map<String, Object>>();
Map<String, Object> task1 = new LinkedHashMap<String, Object>();
task1.put("dataId", UUID.randomUUID().toString());
/**
 * Enter the text to be moderated. The text can be up to 10,000 characters in length.
 */
task1.put("content", "test content");
tasks.add(task1);
JSONObject data = new JSONObject();

/**
 * Set the scenes parameter to antispam.
 */
data.put("scenes", Arrays.asList("antispam"));
data.put("tasks", tasks);
System.out.println(JSON.toJSONString(data, true));
textScanRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", FormatType.JSON);
// You must set the connection timeout and read timeout.
textScanRequest.setConnectTimeout(3000);
textScanRequest.setReadTimeout(6000);
try {
    HttpResponse httpResponse = client.doAction(textScanRequest);
    if(httpResponse.isSuccess()){
        JSONObject scrResponse = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(scrResponse, true));
        if (200 == scrResponse.getInteger("code")) {

```

```
JSONArray taskResults = scrResponse.getJSONArray("data");
for (Object taskResult : taskResults) {
    if(200 == ((JSONObject)taskResult).getInteger("code")){
        JSONArray sceneResults = ((JSONObject)taskResult).getJSONArray("results");
        for (Object sceneResult : sceneResults) {
            String scene = ((JSONObject)sceneResult).getString("scene");
            String suggestion = ((JSONObject)sceneResult).getString("suggestion");
            // Take a further action on the text based on the values of the scene and suggestion parameters.
            // If the value of the suggestion parameter is pass, no spam is detected. If the value of the suggestion parameter is block, spam is detected. In this case, check the value of the label parameter to obtain the spam category.
            System.out.println("args = [" + scene + "]");
            System.out.println("args = [" + suggestion + "]");
        }
    }else{
        System.out.println("task process fail:" + ((JSONObject)taskResult).getInteger("code"));
    }
} else {
    System.out.println("detect not success. code:" + scrResponse.getInteger("code"));
}
}else{
    System.out.println("response not success. status:" + httpResponse.getStatus());
}
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
```

## Provide feedback on the text anti-spam result

If the text anti-spam result does not meet your expectations, you can call the `TextFeedbackRequest` operation to provide feedback on the text anti-spam result. In the feedback, you can pass a correct label for the text moderated.

The server corrects the text anti-spam result based on your feedback and adds the feedback to a text pattern blacklist or whitelist. When you submit a text pattern next time, the server returns the text anti-spam result based on the label that you passed by calling the `TextFeedbackRequest` operation. For more information about the feedback operation, see [Give feedback on moderation results](#).

Operation	Feature	Region	Description
<code>TextFeedbackRequest</code>	Allows you to provide feedback on the text anti-spam result.	<ul style="list-style-type: none"> <li><i>cn-shanghai: China (Shanghai)</i></li> <li><i>cn-beijing: China (Beijing)</i></li> <li><i>cn-shenzhen: China (Shenzhen)</i></li> <li><i>ap-southeast-1: Singapore</i></li> <li><i>us-west-1: US (Silicon Valley)</i></li> </ul>	This operation uses the feedback provided by you to correct the text anti-spam result generated by the text anti-spam algorithm.

### Sample code

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
import com.aliyun.oss.ClientException;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.TextFeedbackRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

public class Main {

    public static void main(String[] args) throws Exception {
        IClientProfile profile = DefaultProfile
            .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
        DefaultProfile
            .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
        IAcsClient client = new DefaultAcsClient(profile);
        TextFeedbackRequest textFeedbackRequest = new TextFeedbackRequest();
        textFeedbackRequest.setAcceptFormat(FormatType.JSON); // Specify the response format of the
        operation.
        textFeedbackRequest.setHttpContentType(FormatType.JSON);
    }
}
```

```
textFeedbackRequest.setMethod(com.aliyuncs.http.MethodType.POST); // Specify the request method.
textFeedbackRequest.setEncoding("UTF-8");
textFeedbackRequest.setRegionId("cn-shanghai");

JSONObject data = new JSONObject();
data.put("taskId", "txt6z3Na17XbrD4P7QdjzYbk1-1q4sej");
data.put("label", "spam");
System.out.println(JSON.toJSONString(data, true));
textFeedbackRequest.setHttpContent(data.toJSONString().getBytes("UTF-8"), "UTF-8", FormatType.JSON);

// You must set the connection timeout and read timeout.
textFeedbackRequest.setConnectTimeout(3000);
textFeedbackRequest.setReadTimeout(6000);
try {
    HttpResponse httpResponse = client.doAction(textFeedbackRequest);
    if(httpResponse.isSuccess()){
        JSONObject scrResponse = JSON.parseObject(new String(httpResponse.getHttpContent(), "UTF-8"));
        System.out.println(JSON.toJSONString(scrResponse, true));
    }else{
        System.out.println("response not success. status:" + httpResponse.getStatus());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
```

## 3.6. OCR

This topic describes how to use the Java SDK to call the optical character recognition (OCR) operation to detect text in common images and return the results in real time.

### Before you begin

Before you call operations, make the following preparations:

- Create an AccessKey pair for your Alibaba Cloud account. For more information, see [Create an AccessKey pair](#).
- Install Java dependencies. For more information, see [Installation](#).
- (Optional)Download and import the [Extension.Uploader utility class](#) into your project if you submit a local image or a binary image stream for image moderation.

## Submit synchronous OCR tasks

### Description

Operation	Feature	Region	Description
ImageSyncScanRequest	Sends synchronous OCR requests to detect text in images.	<ul style="list-style-type: none"> <li>• <i>cn-shanghai</i>: China (Shanghai)</li> <li>• <i>cn-beijing</i>: China (Beijing)</li> <li>• <i>cn-shenzhen</i>: China (Shenzhen)</li> <li>• <i>ap-southeast-1</i>: Singapore</li> <li>• <i>ap-southeast-5</i>: Indonesia (Jakarta)</li> </ul>	This operation detects text in images after you specify ocr in the scenes parameter.

### Sample code

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20180509.ImageSyncScanRequest;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.*;

public class Main {
```

```
public static void main(String[] args) throws Exception {
    IClientProfile profile = DefaultProfile
        .getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
    DefaultProfile
        .addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
    IAcsClient client = new DefaultAcsClient(profile);

    ImageSyncScanRequest imageSyncScanRequest = new ImageSyncScanRequest();
    // Specify the response format of the operation.
    imageSyncScanRequest.setAcceptFormat(FormatType.JSON);
    // Specify the request method.
    imageSyncScanRequest.setMethod(MethodType.POST);
    imageSyncScanRequest.setEncoding("utf-8");
    // Both HTTP and HTTPS are supported.
    imageSyncScanRequest.setProtocol(ProtocolType.HTTP);

    JSONObject httpBody = new JSONObject();
    /**
     * Specify the moderation scenario.
     * ocr
     */
    httpBody.put("scenes", Arrays.asList("ocr"));

    /**
     * Create one task for each image to be moderated.
     * If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
     * Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
     * The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
     */
    JSONObject task = new JSONObject();
    task.put("dataId", UUID.randomUUID().toString());

    // Specify the image URL.
    task.put("url", "https://xxx.jpg");
    httpBody.put("tasks", Arrays.asList(task));

    imageSyncScanRequest.setHttpContent(org.apache.commons.codec.binary.StringUtils.getBytesUtf8
```

```
imageSyncScanRequest.setContentType(org.apache.commons.codec.binary.StringUtils.getBytesOut
f8(httpBody.toJSONString()), "UTF-8", FormatType.JSON);
/**
 * You must set the connection timeout and read timeout. The timeout period for the server to co
mplete an image moderation request is 10s.
 * If you set the read timeout to a value smaller than 10s, a read timeout error is returned.
 */
imageSyncScanRequest.setConnectTimeout(3000);
imageSyncScanRequest.setReadTimeout(10000);
HttpResponse httpResponse = null;
try {
    httpResponse = client.doAction(imageSyncScanRequest);
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
} catch (Exception e){
    e.printStackTrace();
}

// The results returned after the server receives and processes your request.
if(httpResponse != null && httpResponse.isSuccess()){
    JSONObject scrResponse = JSON.parseObject(org.apache.commons.codec.binary.StringUtils.new
StringUtf8(httpResponse.getHttpContent()));
    System.out.println(JSON.toJSONString(scrResponse));
    int requestCode = scrResponse.getIntValue("code");
    // The moderation result of each image.
    JSONArray taskResults = scrResponse.getJSONArray("data");
    if (200 == requestCode) {
        for (Object taskResult : taskResults) {
            // The moderation result of a single image.
            int taskCode = ((JSONObject)taskResult).getIntValue("code");
            // The moderation result of the image in a moderation scenario. If you have specified multi
ple moderation scenarios, the moderation results of the image in each scenario are returned.
            JSONArray sceneResults = ((JSONObject)taskResult).getJSONArray("results");
            if(200 == taskCode){
                for (Object sceneResult : sceneResults) {
                    String scene = ((JSONObject)sceneResult).getString("scene");
                    String suggestion = ((JSONObject)sceneResult).getString("suggestion");
                    //do something
                    // The text detected in the moderated image.
```

```
        if("review".equals(suggestion) && "ocr".equals(scene)){
            JSONObject idCardInfo = ((JSONObject) sceneResult).getJSONObject("idCardInfo");
            System.out.println(idCardInfo.toJSONString());
        }
    }
} else{
    // A single image failed to be moderated. Analyze the failure based on the actual situation.
    System.out.println("task process fail. task response:" + JSON.toJSONString(taskResult));
}
} else {
    /**
     * Your whole request failed to be processed. Analyze the failure based on the actual situation.
     */
    System.out.println("the whole image scan request failed. response:" + JSON.toJSONString(screenResponse));
}
}
}
```

## 3.7. Custom text library

This topic provides the sample code to help you use the Java SDK to manage custom text libraries. You can use the Java SDK to customize a text library and manage the text in the library to meet the personalized requirements in the text anti-spam scenario. Based on the text type, text libraries are classified into term libraries and text pattern libraries. Based on the management purposes, text libraries are classified into whitelists, blacklists, and review lists.

### Before you begin

Before you call operations, make the following preparations:

- Create an AccessKey pair for your Alibaba Cloud account. For more information, see [Create an AccessKey pair](#).
- Install Java dependencies. For more information, see [Installation](#).
- (Optional) Download and import the [Extension.Uploader utility class](#) into your project if you submit a local image or a binary image stream for image moderation.

### Query text libraries

- Query term libraries



```
DescribeKeywordLibRequest describeKeywordLibRequest = new DescribeKeywordLibRequest();
describeKeywordLibRequest.setServiceModule("open_api");

try {
    // This method lists all text libraries, including term libraries and text pattern libraries for text anti-spam, term libraries for detecting ads in images, and term libraries for audio anti-spam.
    DescribeKeywordLibResponse describeKeywordLibResponse = client.getAcsResponse(describeKeywordLibRequest);
    System.out.println(JSON.toJSONString(describeKeywordLibResponse));

    // List text libraries configured for text anti-spam.
    List<DescribeKeywordLibResponse.KeywordLib> allLibs = describeKeywordLibResponse.getKeywordLibList();
    List<DescribeKeywordLibResponse.KeywordLib> textAntispamKeywordLibs = new ArrayList<DescribeKeywordLibResponse.KeywordLib>();
    for (DescribeKeywordLibResponse.KeywordLib keywordLib : allLibs) {
        String libType = keywordLib.getLibType();
        String resourceType = keywordLib.getResourceType();
        String source = keywordLib.getSource();
        // List the term libraries that are customized for text anti-spam.
        if ("textKeyword".equals(libType) && "TEXT".equals(resourceType) && "MANUAL".equals(source)) {
            textAntispamKeywordLibs.add(keywordLib);
        }
        // List the term libraries that are automatically created for text anti-spam based on your feedback.
        if ("textKeyword".equals(libType) && "TEXT".equals(resourceType) && "FEEDBACK".equals(source)) {
            textAntispamKeywordLibs.add(keywordLib);
        }
    }

    System.out.println(JSON.toJSONString(textAntispamKeywordLibs));
} catch (ClientException e) {
    e.printStackTrace();
}
```

- Query text pattern libraries, including those that are customized or automatically created based on your feedback

```
DescribeKeywordLibRequest describeKeywordLibRequest = new DescribeKeywordLibRequest();
describeKeywordLibRequest.setServiceModule("open_api");

try {
    // This method lists all text libraries, including term libraries and text pattern libraries for text anti-spam, term libraries for detecting ads in images, and term libraries for audio anti-spam.
    DescribeKeywordLibResponse describeKeywordLibResponse = client.getAcsResponse(describeKeywordLibRequest);
    System.out.println(JSON.toJSONString(describeKeywordLibResponse));

    // List text pattern libraries.
    List<DescribeKeywordLibResponse.KeywordLib> allLibs = describeKeywordLibResponse.getKeywordLibList();
    List<DescribeKeywordLibResponse.KeywordLib> similarTextLibs = new ArrayList<DescribeKeywordLibResponse.KeywordLib>();
    for (DescribeKeywordLibResponse.KeywordLib keywordLib : allLibs) {
        String libType = keywordLib.getLibType();
        String resourceType = keywordLib.getResourceType();
        String source = keywordLib.getSource();
        // List the text pattern libraries that are customized for text anti-spam.
        if("similarText".equals(libType) && "TEXT".equals(resourceType) && "MANUAL".equals(source)) {
            similarTextLibs.add(keywordLib);
        }

        // List the text pattern libraries that are automatically created for text anti-spam based on your feedback.
        if("similarText".equals(libType) && "TEXT".equals(resourceType) && "FEEDBACK".equals(source)) {
            similarTextLibs.add(keywordLib);
        }
    }

    System.out.println(JSON.toJSONString(similarTextLibs));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Create a text library

- Create a term library

```
CreateKeywordLibRequest createKeywordLibRequest = new CreateKeywordLibRequest();
    createKeywordLibRequest.setServiceModule("open_api");
    createKeywordLibRequest.setName("Term library for testing");
    // Specify that the new library be used for text anti-spam.
    createKeywordLibRequest.setResourceType("TEXT");
    // Specify that the new library be a term library.
    createKeywordLibRequest.setLibType("textKeyword");
    // You can only create a blacklist as the new term library.
    createKeywordLibRequest.setCategory("BLACK");

    try {
        CreateKeywordLibResponse describeKeywordLibResponse = client.getAcsResponse(createKe
ywordLibRequest);
        System.out.println(JSON.toJSONString(describeKeywordLibResponse));
    } catch (ClientException e) {
        e.printStackTrace();
    }
```

- Create a text pattern library

```
CreateKeywordLibRequest createKeywordLibRequest = new CreateKeywordLibRequest();
    createKeywordLibRequest.setServiceModule("open_api");
    createKeywordLibRequest.setName("Text pattern library for testing");
    // Specify that the new library be used for text anti-spam.
    createKeywordLibRequest.setResourceType("TEXT");
    // Specify that the new library be a text pattern library.
    createKeywordLibRequest.setLibType("similarText");
    // You can create a blacklist, whitelist, or review list as the new text pattern library.
    createKeywordLibRequest.setCategory("BLACK");

    try {
        CreateKeywordLibResponse describeKeywordLibResponse = client.getAcsResponse(createKe
ywordLibRequest);
        System.out.println(JSON.toJSONString(describeKeywordLibResponse));
    } catch (ClientException e) {
        e.printStackTrace();
    }
```


## Modify a text library

Use the following code to modify the Name and BizTypes parameters of a text library:

```
UpdateKeywordLibRequest updateKeywordLibRequest = new UpdateKeywordLibRequest();
// Enter the ID of the text library to modify.
updateKeywordLibRequest.setId(2693);
// Enter the new name of the text library.
updateKeywordLibRequest.setName("New name of the text library");
// Set a new value for the BizTypes parameter.
updateKeywordLibRequest.setBizTypes(JSON.toJSONString(Arrays.asList("comment", "title")));

try {
    UpdateKeywordLibResponse updateKeywordLibResponse = client.getAcsResponse(updateKey
wordLibRequest);
    System.out.println(JSON.toJSONString(updateKeywordLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Delete a text library

 **Notice** After you delete a text library, text in the text library is also deleted. You cannot delete a feedback-based text library.

```
DeleteKeywordLibRequest deleteKeywordLibRequest = new DeleteKeywordLibRequest();
// Enter the ID of the text library to delete.
deleteKeywordLibRequest.setId(3353);

try {
    DeleteKeywordLibResponse deleteKeywordLibResponse = client.getAcsResponse(deleteKeywo
rdLibRequest);
    System.out.println(JSON.toJSONString(deleteKeywordLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Search for text in a text library

By default, all text in the text library is returned by page. If you set the Keyword parameter, all text that contains the specified term is returned in fuzzy mode.

```
DescribeKeywordRequest describeKeywordRequest = new DescribeKeywordRequest();
// Enter the ID of the text library in which you want to search for text.
describeKeywordRequest.setKeywordLibId(2693);
describeKeywordRequest.setPageSize(10);
describeKeywordRequest.setCurrentPage(1);
// The Keyword parameter is optional and is used for fuzzy search.
describeKeywordRequest.setKeyword("You");
try {
    DescribeKeywordResponse describeKeywordResponse = client.getAcsResponse(describeKeywordRequest);
    System.out.println(JSON.toJSONString(describeKeywordResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Add text

```
CreateKeywordRequest createKeywordRequest = new CreateKeywordRequest();
// Enter the ID of the text library to which you want to add text.
createKeywordRequest.setKeywordLibId(2693);
// Enter the text to be added.
createKeywordRequest.setKeywords(JSON.toJSONString(Arrays.asList("How do you do", "Perfect")));
try {
    CreateKeywordResponse createKeywordResponse = client.getAcsResponse(createKeywordRequest);
    System.out.println(JSON.toJSONString(createKeywordResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Delete text

```
DeleteKeywordRequest deleteKeywordRequest = new DeleteKeywordRequest();
    // Enter the ID of the text library from which you want to delete text.
    deleteKeywordRequest.setKeywordLibId(String.valueOf(2693));
    // Enter the IDs of the text to delete.
    deleteKeywordRequest.setIds(JSON.toJSONString(Arrays.asList(1, 2)));

    try {
        DeleteKeywordResponse deleteKeywordResponse = client.getAcsResponse(deleteKeywordRequest);
        System.out.println(JSON.toJSONString(deleteKeywordResponse));
    } catch (ClientException e) {
        e.printStackTrace();
    }
```

## 3.8. Custom image library

This topic describes the operations provided by the Java SDK to manage custom image libraries. You can call these operations to customize image samples based on personalized content management requirements to intelligently detect pornography, terrorist content, and ads in images and videos.

### Query custom image libraries

Use the following code to obtain the list of image libraries, including custom image libraries and feedback-based image libraries:

```
DescribeImageLibRequest describeImageLibRequest = new DescribeImageLibRequest();
describeImageLibRequest.setServiceModule("open_api");

try {
    // List all image libraries, including custom image libraries and feedback-based image libraries.
    DescribeImageLibResponse describeImageLibResponse = client.getAcsResponse(describeImage
LibRequest);
    System.out.println(JSON.toJSONString(describeImageLibResponse));

    List<DescribeImageLibResponse.ImageLib> allLibs = describeImageLibResponse.getImageLibList();
    List<DescribeImageLibResponse.ImageLib> customImageLibs = new ArrayList<DescribeImageLib
Response.ImageLib>();
    for (DescribeImageLibResponse.ImageLib imageLib : allLibs) {
        String source = imageLib.getSource();
        // List custom image libraries.
        if("MANUAL".equals(source)) {
            customImageLibs.add(imageLib);
        }

        // List feedback-based image libraries.
        if("FEEDBACK".equals(source)) {
            customImageLibs.add(imageLib);
        }
    }

    System.out.println(JSON.toJSONString(customImageLibs));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Create a custom image library

Use the following code to create a custom image library:

 **Note** Set parameters based on the moderation scenario.

```
CreateImageLibRequest createImageLibRequest = new CreateImageLibRequest();
createImageLibRequest.setServiceModule("open_api");
createImageLibRequest.setName("Pornography detection blacklist");
createImageLibRequest.setScene("PORN");
createImageLibRequest.setCategory("BLACK");

try {
    CreateImageLibResponse createImageLibResponse = client.getAcsResponse(createImageLibRequest);
    System.out.println(JSON.toJSONString(createImageLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Modify a custom image library

Use the following code to modify the Name and BizTypes parameters of a custom image library:

```
UpdateImageLibRequest updateImageLibRequest = new UpdateImageLibRequest();
// Enter the ID of the custom image library.
updateImageLibRequest.setId(12345);
updateImageLibRequest.setName("New name of the custom image library");
updateImageLibRequest.setBizTypes(JSON.toJSONString(Arrays.asList("comment")));
updateImageLibRequest.setCategory("WHITE");
updateImageLibRequest.setScene("PORN");

try {
    UpdateImageLibResponse updateImageLibResponse = client.getAcsResponse(updateImageLibRequest);
    System.out.println(JSON.toJSONString(updateImageLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Delete a custom image library

Use the following code to delete a custom image library:

 **Note** After you delete a custom image library, all images in the library are also deleted.



```
DeleteImageLibRequest deleteImageLibRequest = new DeleteImageLibRequest();
// Enter the ID of the custom image library.
deleteImageLibRequest.setId(12345);
try {
    DeleteImageLibResponse deleteImageLibResponse = client.getAcsResponse(deleteImageLibRequest);
    System.out.println(JSON.toJSONString(deleteImageLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Query images in a custom image library

Use the following code to obtain the list of all images in a custom image library:

```
DescribeImageFromLibRequest describeImageFromLibRequest = new DescribeImageFromLibRequest();
describeImageFromLibRequest.setImageLibId(1519);
describeImageFromLibRequest.setPageSize(20);
describeImageFromLibRequest.setCurrentPage(1);
try {
    DescribeImageFromLibResponse describeImageFromLibResponse = client.getAcsResponse(describeImageFromLibRequest);
    System.out.println(JSON.toJSONString(describeImageFromLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

## Add images to a custom image library

Use the following code to add images to a custom image library. Follow these steps to add images:

1. Obtain the image upload credential.
2. Upload images to the custom image library.
3. Submit the information about the library to which images are uploaded and the image path information to the server.

```
// Obtain the image upload credential.
DescribeUploadInfoRequest describeUploadInfoRequest = new DescribeUploadInfoRequest();
describeUploadInfoRequest.setBiz("customImageLib");
DescribeUploadInfoResponse describeUploadInfoResponse = null;
try {
    describeUploadInfoResponse = client.getAcResponse(describeUploadInfoRequest);
    System.out.println(JSON.toJSONString(describeUploadInfoResponse));
} catch (ClientException e) {
    e.printStackTrace();
}

// Upload images.
CustomLibUploader customLibUploader = new CustomLibUploader();
String object = null;
try {
    object = customLibUploader.uploadFile(describeUploadInfoResponse.getHost(), describeUploadInfoResponse.getFolder(), describeUploadInfoResponse.getAccessid(),
        describeUploadInfoResponse.getPolicy(), describeUploadInfoResponse.getSignature(),
        "/Users/liuhai.lh/Desktop/a.jpg");
} catch (Exception e) {
    e.printStackTrace();
}

if(org.apache.commons.lang.StringUtils.isNotBlank(object)){
    UploadImageToLibRequest imageToLibRequest = new UploadImageToLibRequest();
    imageToLibRequest.setImageLibId(1519);
    imageToLibRequest.setImages(JSON.toJSONString(Arrays.asList(object)));

    try {
        UploadImageToLibResponse uploadImageToLibResponse = client.getAcResponse(imageToLibRequest);
        System.out.println(JSON.toJSONString(uploadImageToLibResponse));
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
```

## Delete custom images from a custom image library

Use the following code to delete multiple custom images from a custom image library:

```
DeletelImageFromLibRequest deletelImageFromLibRequest = new DeletelImageFromLibRequest();
deletelImageFromLibRequest.setIds(JSON.toJSONString(Arrays.asList(669310)));
try {
    DeletelImageFromLibResponse deletelImageFromLibResponse = client.getAcsResponse(deletelImageF
romLibRequest);
    System.out.println(JSON.toJSONString(deletelImageFromLibResponse));
} catch (ClientException e) {
    e.printStackTrace();
}
```

# 4. Python SDK

## 4.1. Installation

Before you use Content Moderation SDK for Python, you must install pip and then use pip to install Python dependencies.

### Prerequisites

- Python 2.x or Python 3.x is used.
- The python-pip dependency library is installed on the source server.

Run the following commands to install the python3-pip dependency library for specific Linux distributions:

- CentOS and Red Hat Enterprise Linux:

```
yum -y install python3-pip
```

- Ubuntu and Debian:

```
apt-get -y install python3-pip
```

- OpenSUSE and SUSE:

```
zypper -n install python3-pip
```

### Install the SDK

- If you are using Python 2.x, run the following commands to install the SDK core library:

```
sudo pip install aliyun-python-sdk-core==2.13.10
pip install -v aliyun-python-sdk-green==3.6.1 // Install the SDK of the specified version.
pip install oss2 // Install the Object Storage Service (OSS) dependency.
```

- If you are using Python 3.x, run the following commands to install the SDK core library:

```
sudo pip install aliyun-python-sdk-core-v3==2.13.10
pip install -v aliyun-python-sdk-green==3.6.1 // Install the SDK of the specified version.
pip install oss2 // Install the OSS dependency.
```

- Download and import the [Extension.Uploader utility class](#) into your project if you submit a local or binary file for content moderation.

## 4.2. Initialization

To use the Python SDK to initiate an API call, initialize a client instance and modify the client configurations as needed.

### Precautions

When creating a client instance, you must specify the region and the AccessKey ID and AccessKey secret of your Alibaba Cloud account.

- For more information about the supported regions, see [Request structure](#).
- For more information about how to obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account, see [Create an AccessKey](#).

## Create a client instance to moderate images, audio, videos, and text

The following regions are supported:

- cn-shanghai: China (Shanghai)
- cn-beijing: China (Beijing)
- ap-southeast-1: Singapore
- us-west-1: US (Silicon Valley)

Use the following code to create a client instance to moderate images, audio, videos, and text:

```
from aliynsdckore import client

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account. You can add the AccessK
ey ID and AccessKey secret to the profile or directly enter them in plaintext in the code.

clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
```

## Create a client instance to manage custom image libraries, term libraries, and text pattern libraries

The supported region is cn-shanghai. The server can automatically synchronize data if you initiate a request in a region other than cn-shanghai.

Use the following code to create a client instance to manage custom image libraries, term libraries, and text pattern libraries:

```
from aliynsdckore import client

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account. You can add the AccessK
ey ID and AccessKey secret to the profile or directly enter them in plaintext in the code.

clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
```

# 4.3. Image moderation

This topic describes how to use the Python SDK to moderate images for risky content.

- The Python SDK supports both synchronous and asynchronous image moderation. If you use synchronous image moderation, the moderation results are returned in real time. If you use asynchronous image moderation, you must poll the moderation results or configure a callback notification to receive the moderation results.
- You can submit the URL of an online image, the URL of a local image, or a binary image stream for image moderation.

## Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install Python dependencies. For more information, see [Installation](#).

## (Recommended) Submit synchronous image moderation tasks

### Operation description


### Sample code

- Submit the URL of an online image for image moderation

```
#coding=utf-8
# The ImageSyncScanRequest operation returns image moderation results in real time.
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import ImageSyncScanRequest
from aliyunsdkgreenextension.request.extension import HttpContentHelper
import json
import uuid

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret","cn-shanghai")
request = ImageSyncScanRequest.ImageSyncScanRequest()
request.set_accept_format('JSON')
task = {
    "dataId": str(uuid.uuid1()),
    "url": "https://test1.jpg"
}

# Create one task for each image to be moderated.
# If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
# Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
# The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
```

```

# For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
request.set_content(HttpContentHelper.toValue({"tasks": [task], "scenes": ["porn"]})))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            for sceneResult in sceneResults:
                scene = sceneResult["scene"]
                suggestion = sceneResult["suggestion"]
                # Take a further action on the image based on the values of the scene and suggestion parameters.
                # do something
                print(suggestion)
                print(scene)

```

- Submit the URL of a local image for image moderation

```

#coding=utf-8
# The ImageSyncScanRequest operation returns image moderation results in real time.
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import ImageSyncScanRequest
from aliyunsdkgreen.request.extension import ClientUploader
from aliyunsdkgreenextension.request.extension import HttpContentHelper
import json
import uuid
import sys

# Set the default encoding to UTF-8 to support a local path containing Chinese characters.
# Add the following code snippet if you use Python 2. If you use Python 3, remove it.
if sys.version_info[0] == 2:
    reload(sys)
    sys.setdefaultencoding('utf-8')

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = ImageSyncScanRequest.ImageSyncScanRequest()

```

```
request.set_accept_format('JSON')

# Use the path of your local image.
# Upload the local image to the server.
uploader = ClientUploader.getImageClientUploader(clt)
url = uploader.uploadFile('d:/test/test.jpg')

# Set the url parameter to the image URL submitted to the server.
task = {"dataId": str(uuid.uuid1()),
        "url":url
        }

# Create one task for each image to be moderated.
# If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
# Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
# The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
# For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
request.set_content(Helper.toValue({"tasks": [task], "scenes": ["porn"]})))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            for sceneResult in sceneResults:
                # Take a further action on the image based on the values of the scene and suggestion parameters.
                # do something
                scene = sceneResult["scene"]
                suggestion = sceneResult["suggestion"]
```



- Submit a binary image stream for image moderation

```
#coding=utf-8
# The ImageSyncScanRequest operation returns image moderation results in real time.
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import ImageSyncScanRequest
from aliyunsdkgreen.request.extension import ClientUploader
from aliyunsdkgreenextension.request.extension import HttpContentHelper
import json
import uuid
import sys

# Set the default encoding to UTF-8 to support a local path containing Chinese characters.
# Add the following code snippet if you use Python 2. If you use Python 3, remove it.
if sys.version_info[0] == 2:
    reload(sys)
    sys.setdefaultencoding('utf-8')

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret","cn-shanghai")
request = ImageSyncScanRequest.ImageSyncScanRequest()
request.set_accept_format('JSON')

# Simulate binary data for moderation.
# Read and convert a local image to binary data. Use the path of your local image.
f = open('d:/test/test.jpg','rb+')
imageBytes = f.read()
f.close()

# Upload the binary stream to the server.
uploader = ClientUploader.getImageClientUploader(clt)
url = uploader.uploadBytes(imageBytes)

# Set the url parameter to the image URL submitted to the server.
task = {
    "dataId": str(uuid.uuid1()),
    "url":url
}

# Create one task for each image to be moderated.
# If you moderate multiple images in a request, the total response time that the server spends proc
```

```
essing the request starts from when the request is initiated to when the last image is moderated.
# Generally, the average response time of moderating multiple images in a request is longer than th
at of moderating a single image. The more images you submit at a time, the higher the probability th
at the average response time will be extended.
# The sample code uses a single image as an example. If you want to moderate multiple images at a
time, create one task for each image to be moderated.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple images at a time and specify multiple moderation sc
enarios for each image. The expenses of all scenarios are calculated separately and summed up.
# For example, if you moderate two images for both pornography and terrorist content, you are cha
rged for moderating two images for pornography and two images for terrorist content.
request.set_content(Helper.toValue({"tasks": [task], "scenes": ["porn"]})))
response = client.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            for sceneResult in sceneResults:
                # Take a further action on the image based on the values of the scene and suggestion par
ameters.
                # do something
                scene = sceneResult["scene"]
                suggestion = sceneResult["suggestion"]
```

## Submit asynchronous image moderation tasks

The following section describes how to use the `ImageAsyncScanRequest` operation of the Python SDK to moderate images for risky content. You can send an asynchronous request to submit an image moderation task. When submitting the request, you can configure a callback notification to receive the moderation results. Alternatively, you can call the `ImageAsyncScanResultsRequest` operation to poll the moderation results.

As with synchronous image moderation, you can submit the URL of an online image, the URL of a local image, or a binary image stream for asynchronous image moderation. The following sample code uses the URL of an online image as an example.

### Sample code

```
# coding=utf-8
# Call the ImageAsyncScanRequest operation to moderate images. Save the task ID returned by the op
eration and use it to poll the moderation results.
from aliyunsdkcore import client
```

```
from aliysdkcore.profile import region_provider
from aliysdkgreen.request.v20180509 import ImageAsyncScanRequest
from aliysdkgreenextension.request.extension import HttpContentHelper
import json
import uuid

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient('Your AccessKey ID', 'Your AccessKey secret', 'cn-shanghai')
region_provider.modify_point('Green', 'cn-shanghai', 'green.cn-shanghai.aliyuncs.com')
request = ImageAsyncScanRequest.ImageAsyncScanRequest()
request.set_accept_format('JSON')

task1 = {
    "dataId": str(uuid.uuid1()),
    "url": "http://xxx.jpg"
}

# Create one task for each image to be moderated.
# If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
# Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
# The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
# For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
request.set_content(HttpContentHelper.toValue({"tasks": [task1], "scenes": ["porn"]}))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if(200 == taskResult["code"]):
            taskId = taskResult["taskId"]
            # Save the task ID, which is used to poll the moderation results at the specified interval. For more information, see the description of the ImageAsyncScanResultsRequest operation.
```

```
print(taskId)
```

## Query the results of asynchronous image moderation

### Operation description


### Sample code

```
# coding=utf-8
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdkgreen.request.v20180509 import ImageAsyncScanResultsRequest
from aliyunsdkgreenextension.request.extension import HttpContentHelper
import json

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
region_provider.modify_point('Green', 'cn-shanghai', 'green.cn-shanghai.aliyuncs.com')
request = ImageAsyncScanResultsRequest.ImageAsyncScanResultsRequest()
request.set_accept_format('JSON')

# Use task IDs to query the result of each image moderation task.
taskIds = ["img5s0$Zsssss7RYuz4Yyhhe-1q51iZ"]
request.set_content(HttpContentHelper.toValue(taskIds))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            for sceneResult in sceneResults:
                # Take a further action on the image based on the values of the scene and suggestion parameters.
                # do something
                scene = sceneResult["scene"]
                suggestion = sceneResult["suggestion"]
```

## 4.4. Video moderation

This topic describes how to use the Python SDK to moderate videos for risky content. You can moderate both images and audio in videos.

- The Python SDK supports both synchronous and asynchronous video moderation. If you use synchronous video moderation, you can only submit a sequence of image frames captured from a video for moderation. If you use asynchronous video moderation, you can submit either a video or a sequence of image frames captured from the video for moderation. We recommend that you use asynchronous video moderation.
- You can submit the URL of an online video, the URL of a local video, a binary video stream, or a video live stream for video moderation.

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install Python dependencies. For more information, see [Installation](#).

### (Recommended) Submit asynchronous video moderation tasks

#### Operation description

Operation	Feature	Region	Description
VideoAsyncScanRequest	Sends asynchronous requests to moderate videos for risky content.	<ul style="list-style-type: none"> <li>• <i>cn-shanghai: China (Shanghai)</i></li> <li>• <i>cn-beijing: China (Beijing)</i></li> <li>• <i>cn-shenzhen: China (Shenzhen)</i></li> <li>• <i>ap-southeast-1: Singapore</i></li> <li>• <i>us-west-1: US (Silicon Valley)</i></li> </ul>	This operation moderates videos for risky content in multiple moderation scenarios, including pornography, terrorist content, ad, undesirable scene, and logo detection.

#### Sample code

- Submit the URL of an online video for video moderation

```

#coding=utf-8
# Call the VideoAsyncScanRequest operation to moderate videos.
from aliynsdkcore import client
from aliynsdkgreen.request.v20180509 import VideoAsyncScanRequest
from aliynsdkgreen.request.extension import HttpContentHelper
import json
import uuid

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoAsyncScanRequest.VideoAsyncScanRequest()
request.set_accept_format('JSON')

task = {"dataId": str(uuid.uuid1()),
        "url": "https://xxx/xxx.mp4"
        }
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want
to submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation sce
narios for each video. The system charges you based on the number of frames captured from each v
ideo and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the
total expense is calculated based on the following formula: Total expense = Number of frames capt
ured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terr
orist content detection scenario).
request.set_content(HttpContentHelper.toValue({"tasks": [task], "scenes": ["terrorism"]})))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        # Save the task ID, which is used to poll the moderation results at the specified interval.
        print(taskResult["taskId"])

```

- Submit the URL of a local video for video moderation

```

#coding=utf-8
# Call the VideoAsyncScanRequest operation to moderate videos.
from aliynsdkcore import client

```

```
from auyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import VideoAsyncScanRequest
from aliyunsdkgreen.request.extension import HttpContentHelper
from aliyunsdkgreen.request.extension import ClientUploader
import json
import uuid

# Set the default encoding to UTF-8 to support a local path containing Chinese characters.
# Add the following code snippet if you use Python 2. If you use Python 3, remove it.
if sys.version_info[0] == 2:
    reload(sys)
    sys.setdefaultencoding('utf-8')

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoAsyncScanRequest.VideoAsyncScanRequest()
request.set_accept_format('JSON')
# Use the path of your local video.
# Upload the local video to the server.
uploader = ClientUploader.getVideoClientUploader(clt)
url = uploader.uploadFile('d:/Terrorist content 1.mp4')

task = {"dataId": str(uuid.uuid1()),
        "url": url
        }
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want
to submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation sce
narios for each video. The system charges you based on the number of frames captured from each v
ideo and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the
total expense is calculated based on the following formula: Total expense = Number of frames capt
ured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terr
orist content detection scenario).
request.set_content(HttpContentHelper.toValue({"tasks": [task], "scenes": ["terrorism"]}))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
```

```
taskResults = result["data"]
for taskResult in taskResults:
    # Save the task ID, which is used to poll the moderation results at the specified interval.
    print(taskResult["taskId"])
```

- **Submit a binary video stream for video moderation**

```
#coding=utf-8
# Call the VideoAsyncScanRequest operation to moderate videos.
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import VideoAsyncScanRequest
from aliyunsdkgreen.request.extension import HttpContentHelper
from aliyunsdkgreen.request.extension import ClientUploader
import json
import uuid

# Set the default encoding to UTF-8 to support a local path containing Chinese characters.
# Add the following code snippet if you use Python 2. If you use Python 3, remove it.
if sys.version_info[0] == 2:
    reload(sys)
    sys.setdefaultencoding('utf-8')

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoAsyncScanRequest.VideoAsyncScanRequest()
request.set_accept_format('JSON')
# Simulate binary data for moderation.
# Read and convert a local video to binary data. Use the path of your local video.
f = open('d:/Terrorist content 1.mp4', "rb+")
videoBytes = f.read()
f.close()

# Upload the binary stream to the server.
uploader = ClientUploader.getVideoClientUploader(clt)
url = uploader.uploadBytes(videoBytes)

task = {"dataId": str(uuid.uuid1()),
        "url": url
        }
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want
to submit more than one task in a request, submit a ticket.
```



```
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios for each video. The system charges you based on the number of frames captured from each video and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames captured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario).
request.set_content(Helper.toValue({"tasks": [task], "scenes": ["terrorism"]}))
response = client.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        # Save the task ID, which is used to poll the moderation results at the specified interval.
        print(taskResult["taskId"])
```

- Submit a video live stream for video moderation

```
#coding=utf-8
# Call the VideoAsyncScanRequest operation to moderate videos.
from aliynsdcore import client
from aliynsdgreen.request.v20180509 import VideoAsyncScanRequest
from aliynsdgreen.request.extension import HttpContentHelper
import json
import uuid

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoAsyncScanRequest.VideoAsyncScanRequest()
request.set_accept_format('JSON')

# Set the url parameter to the URL of your live stream.
task = {
    "dataId": str(uuid.uuid1()),
    "url": "http://xxx/xxx.flv"
}
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want
to submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation sce
narios for each video. The system charges you based on the number of frames captured from each v
ideo and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the
total expense is calculated based on the following formula: Total expense = Number of frames capt
ured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terr
orist content detection scenario).
request.set_content(HttpContentHelper.toValue({"tasks": [task], "scenes": ["terrorism"], "live": True
}))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        # Save the task ID, which is used to poll the moderation results at the specified interval.
        print(taskResult["taskId"])
```

- Submit a video live stream to moderate both the video images and audio

```
#coding=utf-8
# Call the VideoAsyncScanRequest operation to moderate videos.
from aliynsdcore import client
from aliynsdgreen.request.v20180509 import VideoAsyncScanRequest
from aliynsdgreen.request.extension import HttpContentHelper
import json
import uuid

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoAsyncScanRequest.VideoAsyncScanRequest()
request.set_accept_format('JSON')

# Set the url parameter to the URL of your live stream.
task = {
    "dataId": str(uuid.uuid1()),
    "url": "http://xxx/xxx.flv"
}
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want
to submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation sce
narios for each video. The system charges you based on the number of frames captured from each v
ideo and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the
total expense is calculated based on the following formula: Total expense = Number of frames capt
ured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terr
orist content detection scenario).
# If you moderate both images and audio in a video, the expense for image moderation is calculated
based on the preceding rule and that for audio moderation is calculated based on the audio duratio
n.
request.set_content(HttpContentHelper.toValue({"tasks": [task], "scenes": ["terrorism"], "live": True
, "audioScenes": ["antispam"]})))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
```

```
# Save the task ID, which is used to poll the moderation results at the specified interval.  
print(taskResult["taskId"])
```

## Query the results of asynchronous video moderation

### Operation description

Operation	Feature	Region	Description
VideoAsyncScanResultsRequest	Queries the results of asynchronous video moderation tasks.	<ul style="list-style-type: none"><li><i>cn-shanghai: China (Shanghai)</i></li><li><i>cn-beijing: China (Beijing)</i></li><li><i>cn-shenzhen: China (Shenzhen)</i></li><li><i>ap-southeast-1: Singapore</i></li><li><i>us-west-1: US (Silicon Valley)</i></li></ul>	This operation queries the video moderation results through polling. We recommend that you configure a callback notification to receive the moderation results.

### Sample code

```
#coding=utf-8
# Call the VideoAsyncScanResultsRequest operation to query the results of asynchronous video mode
ration tasks.
import json
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import VideoAsyncScanResultsRequest
from aliyunsdkgreen.request.extension import HttpContentHelper

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account. You can add the AccessK
ey ID and AccessKey secret to the profile or directly enter them in plaintext in the code.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret",'cn-shanghai')
request = VideoAsyncScanResultsRequest.VideoAsyncScanResultsRequest()
request.set_accept_format('JSON')

# The taskId parameter indicates the list of video moderation task IDs that are used to query the vid
eo moderation results.
taskId = ['vi3pX@vXC94hPnWsss39WOQ9-1q52ZG']
request.set_content(HttpContentHelper.toValue(taskId))
response = clt.do_action_with_exception(request)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        # The results parameter of each task indicates the moderation results of frames captured from a
        video.
        print(taskResult['results'])
```

## Submit synchronous video moderation tasks

### Operation description

Operation	Feature	Region	Description
VideoSyncScanRequest	Sends synchronous requests to moderate videos for risky content.	<ul style="list-style-type: none"> <li><i>cn-shanghai: China (Shanghai)</i></li> <li><i>cn-beijing: China (Beijing)</i></li> <li><i>cn-shenzhen: China (Shenzhen)</i></li> <li><i>ap-southeast-1: Singapore</i></li> <li><i>us-west-1: US (Silicon Valley)</i></li> </ul>	If you use synchronous video moderation, you can only submit a sequence of image frames captured from a video for moderation. We recommend that you use asynchronous video moderation.

Operation	Feature	Region	Description
-----------	---------	--------	-------------

### Sample code

```
#coding=utf-8
# Call the VideoSyncScanRequest operation to moderate a sequence of frames captured from a video.
from aliynsdcore import client
from aliynsdkgreen.request.v20180509 import VideoSyncScanRequest
from aliynsdkgreen.request.extension import HttpContentHelper
import json


# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", "cn-shanghai")
request = VideoSyncScanRequest.VideoSyncScanRequest()
request.set_accept_format('JSON')

task = {
    "frames":[
        {"offset" : 0, "url" : "https://test1.jpg"},
        {"offset" : 2, "url" : "https://test2.jpg"},
        {"offset" : 3, "url" : "https://test3.jpg"}
    ]
}
print(task)
# Create a video moderation task. By default, you can only submit one task in a request. If you want to
submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation scena
rios for each video. The system charges you based on the number of frames captured from each video
and the number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the t
otal expense is calculated based on the following formula: Total expense = Number of frames capture
```

```
d from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist c
ontent detection scenario).
request.set_content(Helper.toValue({"tasks": [task], "scenes": ["porn"]}))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        for result in taskResult["results"]:
            # Take a further action on the video based on the values of the scene and suggestion paramet
            ers.
            print(result['suggestion'])
            print(result['scene'])
```

## 4.5. Text anti-spam

This topic describes how to use the Python SDK to moderate text for spam such as pornography and terrorist content.

 **Note** Currently, the Python SDK only supports synchronous text moderation. You can send a request to moderate one or more pieces of text. The system charges you based on the number of pieces of text moderated. For more information, see [Billing methods](#).

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install Python dependencies. For more information, see [Installation](#).

### Moderate text for spam

Text anti-spam allows you to add custom terms, such as brand terms of competitors. If the text being moderated contains the terms you add, the value of the suggestion parameter returned by the text anti-spam algorithm is block.

You can log on to the [Alibaba Cloud Content Moderation console](#) or call the corresponding operation to add terms. The terms are UTF-8-encoded. For more information about the text anti-spam operation, see [Moderate text for spam](#).

Sample code

```
#coding=utf-8
# Call the TextScanRequest operation.
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdkgreen.request.v20180509 import TextScanRequest
from aliyunsdkgreen.request.extension import HttpContentHelper
import json
import uuid
import datetime

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account. You can add the AccessKey ID and AccessKey secret to the profile or directly enter them in plaintext in the code.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", 'cn-shanghai')
region_provider.modify_point('Green', 'cn-shanghai', 'green.cn-shanghai.aliyuncs.com')
request = TextScanRequest.TextScanRequest()
request.set_accept_format('JSON')
task1 = {"dataId": str(uuid.uuid1()),
        "content": "Text to be moderated",
        "time": datetime.datetime.now().microsecond
        }
# Set the scenes parameter to antispam.
request.set_content(HttpContentHelper.toValue({"tasks": [task1], "scenes": ["antispam"]}))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            for sceneResult in sceneResults:
                scene = sceneResult["scene"]
                suggestion = sceneResult["suggestion"]
                # Take a further action on the text based on the values of the scene and suggestion parameters.
                # do something
```

## Provide feedback on the text anti-spam result



If the text anti-spam result does not meet your expectations, you can call the `TextFeedbackRequest` operation to provide feedback on the text anti-spam result. In the feedback, you can pass a correct label for the text moderated.

The server corrects the text anti-spam result based on your feedback and adds the feedback to a text pattern blacklist or whitelist. When you submit a text pattern next time, the server returns the text anti-spam result based on the label that you passed by calling the `TextFeedbackRequest` operation. For more information about the feedback operation, see [Give feedback on moderation results](#).

### Sample code

```
#coding=utf-8
# Call the TextFeedbackRequest operation.
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdkgreen.request.v20180509 import TextFeedbackRequest
from aliyunsdkgreen.request.extension import HttpContentHelper

# Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret", 'cn-shanghai')
region_provider.modify_point('Green', 'cn-shanghai', 'green.cn-shanghai.aliyuncs.com')
request = TextFeedbackRequest.TextFeedbackRequest()
request.set_accept_format('JSON')

# The taskId parameter indicates the ID of the text moderation task for which you want to provide feedback. The label parameter indicates the actual text anti-spam result.
request.set_content(HttpContentHelper.toValue({"taskId": "txt6z3Na17XbrD4P7QdjzYbk1-1q4sej", "label": "normal"}))
response = clt.do_action_with_exception(request)
print(response)
```

## 4.6. OCR

This topic describes how to use the Python SDK to call the optical character recognition (OCR) operation to detect text in common images and return the results in real time.

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install Python dependencies. For more information, see [Installation](#).

### Submit synchronous OCR tasks

**Description**

Operation	Feature	Region	Description
ImageSyncScanRequest	Sends synchronous OCR requests to detect text in images.	<ul style="list-style-type: none"><li>• <i>cn-shanghai</i>: China (Shanghai)</li><li>• <i>cn-beijing</i>: China (Beijing)</li><li>• <i>cn-shenzhen</i>: China (Shenzhen)</li><li>• <i>ap-southeast-1</i>: Singapore</li><li>• <i>ap-southeast-5</i>: Indonesia (Jakarta)</li></ul>	This operation detects text in images after you specify ocr in the scenes parameter.

**Sample code**

```
#coding=utf-8
# The OCR operation returns OCR results in real time.
from aliyunsdkcore import client
from aliyunsdkgreen.request.v20180509 import ImageSyncScanRequest
from aliyunsdkgreen.request.extension import HttpContentHelper
import json
import uuid

clt = client.AcsClient("Your AccessKey ID", "Your AccessKey secret","cn-shanghai")
request = ImageSyncScanRequest.ImageSyncScanRequest()
request.set_accept_format('JSON')
task = {"dataId": str(uuid.uuid1()),
        "url":"https://xxx/test.jpg"
        }

print(task)
# Create one task for each image to be moderated.
# If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
# Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
# The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
# The OCR expense equals the product of the number of images moderated and the moderation unit price.
request.set_content(HttpContentHelper.toValue({"tasks": [task],
                                              "scenes": ["ocr"]
                                              )))
response = clt.do_action_with_exception(request)
print(response)
result = json.loads(response)
if 200 == result["code"]:
    taskResults = result["data"]
    for taskResult in taskResults:
        if (200 == taskResult["code"]):
            sceneResults = taskResult["results"]
            print(sceneResults)
```

# 5.PHP SDK

## 5.1. Installation

This topic describes how to install PHP dependencies before you can use the PHP SDK.

### Prerequisites

PHP 5.3 or later is used.

### Download the SDKs

1. Click the following links and download the public PHP SDK packages and the PHP SDK package dedicated to Content Moderation:
  - [aliyun-php-sdk-core](#)
  - [aliyun-php-sdk-green](#)
  - [aliyun-oss-php-sdk](#)
2. Create the *aliyuncs* directory in your project and respectively decompress the preceding SDK packages in the directory. The following directories are generated after the decompression:

```
aliyuncs/aliyun-php-sdk-core/...
aliyuncs/aliyun-php-sdk-green/Green/...
aliyuncs/aliyun-oss-php-sdk/...
```

3. Import the PHP SDK into your code. The sample code is as follows:

```
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;
```

## 5.2. Initialization

To use the PHP SDK to initiate an API call, initialize a client instance and modify the client configurations as needed.

### Precautions

When creating a client instance, you must specify the region and the AccessKey ID and AccessKey secret of your Alibaba Cloud account.

- For more information about the supported regions, see [Request structure](#).
- For more information about how to obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account, see [Create an AccessKey](#).

### Create a client instance to moderate images, audio, videos, and text


The following regions are supported:

- cn-shanghai: China (Shanghai)
- cn-beijing: China (Beijing)
- ap-southeast-1: Singapore

- us-west-1: US (Silicon Valley)

Use the following code to create a client instance to moderate images, audio, videos, and text:

```
// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
;
$client = new DefaultAcsClient($iClientProfile);
```

 **Note** You can replace `cn-shanghai` in the preceding code with `cn-beijing`, `ap-southeast-1`, or `us-west-1` to create a client instance in the corresponding region.

## 5.3. Image moderation

This topic describes how to use the PHP SDK to moderate images for risky content.

- The PHP SDK supports both synchronous and asynchronous image moderation. If you use synchronous image moderation, the moderation results are returned in real time. If you use asynchronous image moderation, you must poll the moderation results or configure a callback notification to receive the moderation results.
- You can submit the URL of an online image, the URL of a local image, or a binary image stream for image moderation.

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install PHP dependencies. For more information, see [Installation](#).

### (Recommended) Submit synchronous image moderation tasks

#### Sample code

- Submit the URL of an online image for image moderation

```
<? php

require_once 'aliyuncs/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;
use Green\Request\Extension\ClientUploader;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
```

```
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);

$request = new Green\ImageSyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

$task1 = array('dataId' => uniqid(),
    'url' => 'http://Your image.jpg'
);
// Create one task for each image to be moderated.
// If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
// Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
// The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
// For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn","terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the image based on the values of the scene and suggestion parameters
```

```
parameters.  
        // do something  
    }  
    }else{  
        print_r("task process fail:" + $response->code);  
    }  
    }  
    }else{  
        print_r("detect not success. code:" + $response->code);  
    }  
} catch (Exception $e) {  
    print_r($e);  
}
```

- Submit the URL of a local image for image moderation

```
<? php  
  
require_once 'aliyun-oss-php-sdk/autoload.php';  
include_once 'aliyun-php-sdk-core/Config.php';  
  
use Green\Request\V20180509 as Green;  
use Green\Request\Extension\ClientUploader;  
  
// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.  
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");  
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyun.com");  
$client = new DefaultAcsClient($iClientProfile);  
  
$request = new Green\ImageSyncScanRequest();  
$request->setMethod("POST");  
$request->setAcceptFormat("JSON");  
  
// Upload your local image before moderation.  
$uploader = ClientUploader::getImageClientUploader($client);  
$url = $uploader->uploadFile("d:/1.png");  
$task1 = array('dataId' => uniqid(),  
    'url' => $url  
);  
// Create one task for each image to be moderated.
```

```
// If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
// Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
// The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
// For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn","terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the image based on the values of the scene and suggestion parameters.
                    // do something
                }
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}
```

- Submit a binary image stream for image moderation



```
<? php
require_once 'aliyun/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;
use Green\Request\Extension\ClientUploader;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\ImageSyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");
$uploader = ClientUploader::getImageClientUploader($client);

// Read and convert a local image to binary data and submit the binary data for moderation.
$bytes = file_get_contents("d:/Your image.jpg");
$url = $uploader->uploadBytes($bytes);
$task1 = array('dataId' => uniqid(),
    'url' => $url
);
// Create one task for each image to be moderated.
// If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
// Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.
// The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.
// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple images at a time and specify multiple moderation scenarios for each image. The expenses of all scenarios are calculated separately and summed up.
// For example, if you moderate two images for both pornography and terrorist content, you are charged for moderating two images for pornography and two images for terrorist content.
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
}
```

```
if(200 == $response->code){
    $taskResults = $response->data;
    foreach ($taskResults as $taskResult) {
        if(200 == $taskResult->code){
            $sceneResults = $taskResult->results;
            foreach ($sceneResults as $sceneResult) {
                $scene = $sceneResult->scene;
                $suggestion = $sceneResult->suggestion;
                // Take a further action on the image based on the values of the scene and suggestion p
arameters.
                // do something
                print_r($scene);
                print_r($suggestion);
            }
        }else{
            print_r("task process fail:" + $response->code);
        }
    }
}else{
    print_r("detect not success. code:" + $response->code);
}
} catch (Exception $e) {
    print_r($e);
}
```

## Submit asynchronous image moderation tasks

The following section describes how to use the ImageAsyncScanRequest operation of the PHP SDK to moderate images for risky content. You can send an asynchronous request to submit an image moderation task. When submitting the request, you can configure a callback notification to receive the moderation results. Alternatively, you can call the ImageAsyncScanResultsRequest operation to poll the moderation results.

As with synchronous image moderation, you can submit the URL of an online image, the URL of a local image, or a binary image stream for asynchronous image moderation. The following sample code uses the URL of an online image as an example.

### Sample code

```
<? php
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKev ID", "Your AccessKev secre
```

```

t");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com")
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\ImageAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");
$task1 = array('dataId' => uniqid(),
    'url' => 'http://xxx.jpg'
);

// Valid values of the scenes parameter are porn, terrorism, qrcode, ad, and ocr.
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn","terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $taskId = $taskResult->taskId;
                // Save the task ID, which is used to poll the moderation results at the specified interval. For
                more information, see the description of the ImageAsyncScanResultsRequest operation.
                print_r($taskId);
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}

```

## Query the results of asynchronous image moderation

### Sample code

```
<? php
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\ImageAsyncScanResultsRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");
$request->setContent(json_encode(array('img4QZbr4m3rGh4Fqp$vsHWsK-1q5mP@')));

try {
    $response = $client->getAcsResponse($request);
    print_r($response, JSON_FORCE_OBJECT);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the image based on the values of the scene and suggestion parameters.
                    // do something
                }
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}
```

## 5.4. Video moderation

This topic describes how to use the PHP SDK to moderate videos for risky content. You can moderate both images and audio in videos.

- The PHP SDK supports both synchronous and asynchronous video moderation. If you use synchronous video moderation, you can only submit a sequence of image frames captured from a video for moderation. If you use asynchronous video moderation, you can submit either a video or a sequence of image frames captured from the video for moderation. We recommend that you use asynchronous video moderation.
- You can submit the URL of an online video, the URL of a local video, a binary video stream, or a video live stream for video moderation.

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install PHP dependencies. For more information, see [Installation](#).

### (Recommended) Submit asynchronous video moderation tasks

#### Operation description

Operation	Feature	Region	Description
VideoAsyncScanRequest	Sends asynchronous requests to moderate videos for risky content.	<ul style="list-style-type: none"> <li>• <i>cn-shanghai: China (Shanghai)</i></li> <li>• <i>cn-beijing: China (Beijing)</i></li> <li>• <i>cn-shenzhen: China (Shenzhen)</i></li> <li>• <i>ap-southeast-1: Singapore</i></li> <li>• <i>us-west-1: US (Silicon Valley)</i></li> </ul>	This operation moderates videos for risky content in multiple moderation scenarios, including pornography, terrorist content, ad, undesirable scene, and logo detection.

#### Sample code

- Submit the URL of an online video for video moderation

```
<? php

require_once 'aliyun/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun/aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;

$clientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey sec
```

```
ret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios for each video. The system charges you based on the number of frames captured from each video and the number of moderation scenarios for each video.
// For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames captured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario).
$task1 = array('dataId' => uniqid(),
    'url' => 'http://xxx.mp4'
);
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $taskId = $taskResult->taskId;
                // Save the task ID, which is used to poll the moderation results at the specified interval. For more information, see the description of the VideoAsyncScanResultsRequest operation.
                print_r($taskId);
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}
```

```
}

```

- Submit the URL of a local video for video moderation

```
<? php

require_once 'aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;
use Green\Request\Extension\ClientUploader;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios for each video. The system charges you based on the number of frames captured from each video and the number of moderation scenarios for each video.
// For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames captured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario). // Submit your local video before moderation.
$uploader = ClientUploader::getVideoClientUploader($client);
$url = $uploader->uploadFile("d:/Terrorist content 1.mp4");
$task1 = array('dataId' => uniqid(),
    'url' => $url
);
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
```

```
foreach ($taskResults as $taskResult) {
    if(200 == $taskResult->code){
        $taskId = $taskResult->taskId;
        // Save the task ID, which is used to poll the moderation results at the specified interval. For more information, see the description of the VideoAsyncScanResultsRequest operation.
        print_r($taskId);
    }else{
        print_r("task process fail:" + $response->code);
    }
}
}else{
    print_r("detect not success. code:" + $response->code);
}
} catch (Exception $e) {
    print_r($e);
}
```

- Submit a binary video stream for video moderation

```
<? php

require_once 'aliyun/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun/aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;
use Green\Request\Extension\ClientUploader;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyun.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios for each video. The system charges you based on the number of frames captured from each video and the number of moderation scenarios for each video.
// For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames ca
```



ptured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario).

```

$uploader = ClientUploader::getVideoClientUploader($client);
$bytes = file_get_contents('d:/test.mp4');
$url = $uploader->uploadBytes($bytes);
$task1 = array('dataId' => uniqid(),
    'url' => $url
);
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $taskId = $taskResult->taskId;
                // Save the task ID, which is used to poll the moderation results at the specified interval. For
                // or more information, see the description of the VideoAsyncScanResultsRequest operation.
                print_r($taskId);
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}

```

- Submit a video live stream for video moderation

```

<? php

require_once 'aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey sec

```

```
ret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple video live streams at a time and specify multiple moderation scenarios for each video live stream. The system charges you based on the number of frames captured from each video live stream and the number of moderation scenarios for each video live stream.
// For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames captured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario).
$task1 = array('dataId' => uniqid(),
    'url' => 'http://URL of your video live stream.flv'
);
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"), "live" => true)));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $taskId = $taskResult->taskId;
                // Save the task ID, which is used to poll the moderation results at the specified interval. For more information, see the description of the VideoAsyncScanResultsRequest operation.
                print_r($taskId);
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
```

```

    catch (Exception $e) {
        print_r($e);
    }
}

```

- Submit a video live stream to moderate both the video images and audio

```

<? php

require_once 'aliyuncs/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

# Create a video moderation task. By default, you can only submit one task in a request. If you want
# to submit more than one task in a request, submit a ticket.
# The system charges you based on the moderation scenario that you specify.
# You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios
# for each video. The system charges you based on the number of frames captured from each video and the
# number of moderation scenarios for each video.
# For example, if you moderate two videos for both pornography and terrorist content at a time, the
# total expense is calculated based on the following formula: Total expense = Number of frames captured
# from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist
# content detection scenario).
# If you moderate both images and audio in a video, the expense for image moderation is calculated
# based on the preceding rule and that for audio moderation is calculated based on the audio duration.

$task1 = array('dataId' => uniqid(),
    'url' => 'http://URL of your video live stream.mp4'
);
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"), "audioScenes" => 'antispam')));
try {
    $response = $client->getAcsResponse($request);
}

```

```

print_r($response);
if(200 == $response->code){
    $taskResults = $response->data;
    foreach ($taskResults as $taskResult) {
        if(200 == $taskResult->code){
            $taskId = $taskResult->taskId;
            // Save the task ID, which is used to poll the moderation results at the specified interval. For
            // or more information, see the description of the VideoAsyncScanResultsRequest operation.
            print_r($taskId);
        }else{
            print_r("task process fail:" + $response->code);
        }
    }
}else{
    print_r("detect not success. code:" + $response->code);
}
} catch (Exception $e) {
    print_r($e);
}

```

## Query the results of an asynchronous video moderation

### Operation description

Operation	Feature	Region	Description
VideoAsyncScanResultsRequest	Queries the results of asynchronous video moderation tasks.	<ul style="list-style-type: none"> <li><i>cn-shanghai: China (Shanghai)</i></li> <li><i>cn-beijing: China (Beijing)</i></li> <li><i>cn-shenzhen: China (Shenzhen)</i></li> <li><i>ap-southeast-1: Singapore</i></li> <li><i>us-west-1: US (Silicon Valley)</i></li> </ul>	This operation queries the video moderation results through polling. We recommend that you configure a callback notification to receive the moderation results.

### Sample code

```

<? php
include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secre

```

```
t");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com")
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoAsyncScanResultsRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// Specify the task ID returned after you submit the asynchronous moderation task.
$request->setContent(json_encode(array("testtaskid")));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the video based on the values of the scene and suggestion parameters.
                    // do something
                }
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}
```

## Submit synchronous video moderation tasks

### Operation description

Operation	Feature	Region	Description
VideoSyncScanRequest	Sends synchronous requests to moderate videos for risky content.	<ul style="list-style-type: none"> <li><i>cn-shanghai: China (Shanghai)</i></li> <li><i>cn-beijing: China (Beijing)</i></li> <li><i>cn-shenzhen: China (Shenzhen)</i></li> <li><i>ap-southeast-1: Singapore</i></li> <li><i>us-west-1: US (Silicon Valley)</i></li> </ul>	If you use synchronous video moderation, you can only submit a sequence of image frames captured from a video for moderation. We recommend that you use asynchronous video moderation.

### Sample code

```
<? php

require_once 'aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun-oss-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;


$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyun-oss.com");
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\VideoSyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

// The system charges you based on the moderation scenario that you specify.
// You can send a request to moderate multiple videos at a time and specify multiple moderation scenarios for each video. The system charges you based on the number of frames captured from each video and the number of moderation scenarios for each video.
// For example, if you moderate two videos for both pornography and terrorist content at a time, the total expense is calculated based on the following formula: Total expense = Number of frames captured from the two videos × (Unit price in the pornography detection scenario + Unit price in the terrorist content detection scenario).
$task1 = array('frames' => array(['offset' => '0', 'url'=>'https://test1.jpg'],
    ['offset' => '1', 'url'=>'https://test2.jpg'],
    ['offset' => '2', 'url'=>'https://test3.jpg'] ));
```

```
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("porn", "terrorism"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the video based on the values of the scene and suggestion parameters.
                    // do something
                }
            }else{
                print_r("task process fail:" + $response->code);
            }
        }
    }
} catch (Exception $e) {
    print_r($e);
}
```

## 5.5. Text anti-spam

This topic describes how to use the PHP SDK to moderate text for spam such as pornography and terrorist content.

 **Note** Currently, the PHP SDK only supports synchronous text moderation. You can send a request to moderate one or more pieces of text. The system charges you based on the number of pieces of text moderated. For more information, see [Billing methods](#).

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install PHP dependencies. For more information, see [Installation](#).

### Moderate text for spam

Text anti-spam allows you to add custom terms, such as brand terms of competitors. If the text being moderated contains the terms you add, the value of the suggestion parameter returned by the text anti-spam algorithm is block.

You can log on to the [Alibaba Cloud Content Moderation console](#) or call the corresponding operation to add terms. The terms are UTF-8-encoded. For more information about the text anti-spam operation, see [Moderate text for spam](#).

#### Sample code

```
<? php

include_once 'aliyun/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\TextScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");
$task1 = array('dataId' => uniqid(),
    'content' => 'Text to be moderated'
);

/**
 * Set the scenes parameter to antispam.
 */
$request->setContent(json_encode(array("tasks" => array($task1),
    "scenes" => array("antispam"))));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if(200 == $taskResult->code){
                $sceneResults = $taskResult->results;
                foreach ($sceneResults as $sceneResult) {
                    $scene = $sceneResult->scene;
                    $suggestion = $sceneResult->suggestion;
                    // Take a further action on the text based on the values of the scene and suggestion para
```



```
meters.  
    // do something  
    }  
    }else{  
        print_r("task process fail:" + $response->code);  
    }  
    }  
    }else{  
        print_r("detect not success. code:" + $response->code);  
    }  
} catch (Exception $e) {  
    print_r($e);  
}
```

## Provide feedback on the text anti-spam result

If the text anti-spam result does not meet your expectations, you can call the `TextFeedbackRequest` operation to provide feedback on the text anti-spam result. In the feedback, you can pass a correct label for the text moderated.

The server corrects the text anti-spam result based on your feedback and adds the feedback to a text pattern blacklist or whitelist. When you submit a text pattern next time, the server returns the text anti-spam result based on the label that you passed by calling the `TextFeedbackRequest` operation. For more information about the feedback operation, see [Give feedback on moderation results](#).

### Sample code

```
<? php

include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20180509 as Green;

$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyuncs.com");
;
$client = new DefaultAcsClient($iClientProfile);
$request = new Green\TextFeedbackRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");
$task1 = array('dataId' => uniqid(),
    'content' => 'Text to be moderated'
);

/**
 * Set the scenes parameter to antispam.
 */
$request->setContent(json_encode(array("taskId" => 'test taskId', "content" => "Text to be moderated",
    "label" => 'spam')));
try {
    $response = $client->getAcsResponse($request);
    print_r($response);
} catch (Exception $e) {
    print_r($e);
}
```

## 5.6. OCR

This topic describes how to use the PHP SDK to call the optical character recognition (OCR) operation to detect text in common images and return the results in real time.

### Preparations

Before calling operations, make the following preparations:

1. Create an AccessKey ID and AccessKey secret of your Alibaba Cloud account. For more information, see [Create an AccessKey](#).
2. Install PHP dependencies. For more information, see [Installation](#).

## Submit synchronous OCR tasks

### Description

Operation	Feature	Region	Description
ImageSyncScanRequest	Sends synchronous OCR requests to detect text in images.	<ul style="list-style-type: none"> <li><i>cn-shanghai</i>: China (Shanghai)</li> <li><i>cn-beijing</i>: China (Beijing)</li> <li><i>cn-shenzhen</i>: China (Shenzhen)</li> <li><i>ap-southeast-1</i>: Singapore</li> <li><i>ap-southeast-5</i>: Indonesia (Jakarta)</li> </ul>	This operation detects text in images after you specify ocr in the scenes parameter.

### Sample code

```
<? php

require_once 'aliyun/aliyun-oss-php-sdk/autoload.php';
include_once 'aliyun/aliyun-php-sdk-core/Config.php';

use Green\Request\V20180509 as Green;

// Use the AccessKey ID and AccessKey secret of your Alibaba Cloud account.
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", "Your AccessKey ID", "Your AccessKey secret");
DefaultProfile::addEndpoint("cn-shanghai", "cn-shanghai", "Green", "green.cn-shanghai.aliyun.com");
;
$client = new DefaultAcsClient($iClientProfile);

$request = new Green\ImageSyncScanRequest();
$request->setMethod("POST");
$request->setAcceptFormat("JSON");

$task1 = array('dataId' => uniqid(),
    'url' => 'https://xxx.jpg'
);
// Create one task for each image to be moderated.
// If you moderate multiple images in a request, the total response time that the server spends processing the request starts from when the request is initiated to when the last image is moderated.
// Generally, the average response time of moderating multiple images in a request is longer than that of moderating a single image. The more images you submit at a time, the higher the probability that a
```

t or moderating a single image. The more images you submit at a time, the higher the probability that the average response time will be extended.

// The sample code uses a single image as an example. If you want to moderate multiple images at a time, create one task for each image to be moderated.


// The OCR expense equals the product of the number of images moderated and the moderation unit price.

```
$request->setContent(json_encode(array(
    "tasks" => array($task1),
    "scenes" => array("ocr")
)));
try {
    $response = $client->getAcsResponse($request);
    # print_r($response);
    if(200 == $response->code){
        $taskResults = $response->data;
        foreach ($taskResults as $taskResult) {
            if (200 == $taskResult->code) {
                $sceneResults = $taskResult->results;
                print_r($sceneResults);
            }
        }
    }else{
        print_r("detect not success. code:" + $response->code);
    }
} catch (Exception $e) {
    print_r($e);
}
```

## 6.SDKs in other languages

If the Content Moderation SDKs of Java, PHP 5.3 or later, and Python do not meet your language requirements, we recommend that you directly send an HTTP request to call the corresponding Content Moderation API operation. For more information, see [Request structure](#) and [List of operations by function](#).

It is difficult to develop and maintain SDKs for a wide variety of development languages. We have also listed some content moderation SDKs compiled by third-party developers for your reference. The third-party SDKs are available in the following languages: **C#**, **C++**, **Node.js**, and **Go**.

 **Note** The third-party SDKs are only for your reference. Alibaba Cloud does not provide follow-up maintenance for them. For more information, see [Development preparations](#).

[Click to download the third-party SDKs for C#, C++, Node.js, and Go.](#)

## 7. Sample code download

Download the sample code that contains call examples of Java, PHP, and Python SDKs for reference. For more information, see the "Development preparations" section in [SDK overview](#).

## 8.SDK release notes

This topic describes the release notes of the Java SDK for Content Moderation.

Latest Java SDK version: aliyun-java-sdk-green 3.5.0

Last update time: 08/19/2019

Previous Java SDK version: aliyun-java-sdk-green 3.4.1

For more information, see [Version history](#).

### Release notes

Release date	SDK	Version	Description
09/19/2019	aliyun-java-sdk-green	3.5.0	Added the facial attribute detection operation.
11/21/2018	aliyun-java-sdk-green	3.4.1	<ul style="list-style-type: none"> <li>Supported the moderation of audio streams and audio files.</li> <li>Supported the moderation of local files.</li> <li>Supported the moderation of video live streams.</li> </ul>
09/04/2018	aliyun-java-sdk-green	3.3.0	Added the speaker recognition operation.
05/12/2017	aliyun-java-sdk-green	3.0.0	<p>Re-designed the Java SDK to make it easier to use and more standard.</p> <ul style="list-style-type: none"> <li>Unified the request and response parameters.</li> <li>Supported image pornography detection as a separate image moderation scenario.</li> <li>Supported Graphics Interchange Format (GIF) images and optimized the moderation of long images.</li> </ul>
12/16/2016	aliyun-java-sdk-green	2.6.0	<p>Updates:</p> <ul style="list-style-type: none"> <li>Returned the risk category in the text anti-spam result.</li> <li>Allowed you to specify the term IDs.</li> <li>Supported the recognition of illegal ads in images.</li> <li>Fixed the issue where the celebrity recognition operation cannot resolve face positions.</li> </ul>
10/18/2016	aliyun-java-sdk-green	2.1.0	Updated the Java SDK to version 2.1.0.

---

Release date	SDK	Version	Description
06/21/2016	aliyun-java-sdk-green	1.4.0	Updated the Java SDK to version 1.4.0.
03/08/2016	aliyun-java-sdk-green	1.1.0	Updated the Java SDK to version 1.1.0.
002/24/2016	aliyun-java-sdk-green	1.0.3	Updated the Java SDK to version 1.0.3.
01/18/2016	aliyun-java-sdk-green	1.0.2	Updated the Java SDK to version 1.0.2.
12/22/2015	aliyun-java-sdk-green	1.0.1	Updated the Java SDK to version 1.0.1.
12/16/2015	aliyun-java-sdk-green	1.0.0	Officially published the Java SDK of version 1.0.0.