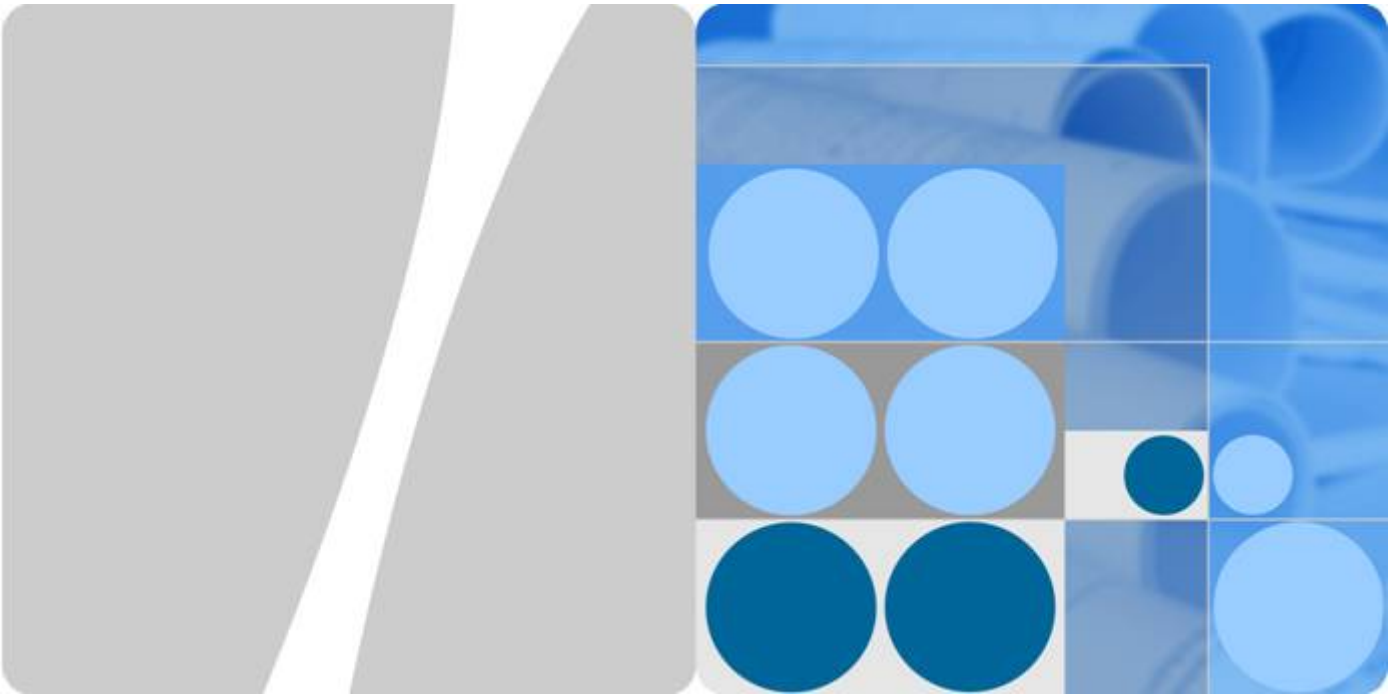


# Content Moderation

# SDK Reference

Issue 07

Date 2020-04-10

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

### Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

### Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

# Contents

1 Applying for a Service ..... 1
2 Obtaining Authentication Information ..... 2
3 Obtaining Moderation SDK ..... 3
4 Preparing the Environment ..... 4
5 Using the SDK (Java) ..... 5
5.1 Preparing a Java Development Environment ..... 5
5.2 Installing Eclipse and Importing SDK Projects ..... 6
5.3 Moderation (Text) ..... 7
5.3.1 Demo Project of Text Moderation ..... 7
5.4 Moderation (Image) ..... 8
5.4.1 Demo Project of Image Moderation ..... 9
5.5 SDK Projects Using Token Authentication ..... 9
6 Using Java SDK (Deprecated) ..... 11
6.1 Demo Project of Distortion Correction ..... 11
6.2 Demo Project of Image Clarity Detect ..... 13
7 Using the Python SDK ..... 15
7.1 Preparing a Python Development Environment ..... 15
7.2 Importing a Python SDK into a Python Environment ..... 16
7.3 SDK Calling Example ..... 17
8 Using the PHP SDK ..... 20
8.1 Preparing a PHP Development Environment ..... 20
8.2 Importing a PHP SDK into a PHP Environment ..... 21
8.3 SDK Calling Example ..... 22
9 Using Node.js SDK ..... 24
9.1 Preparing a Node.js Development Environment ..... 24
9.2 Importing a Node.js SDK into a Node.js Environment ..... 25
9.3 SDK Calling Example ..... 26
10 Change History ..... 28

# 1 Applying for a Service

For details about how to apply for a service, see **Applying for a Service** in the *Content Moderation API Reference*.

# 2 Obtaining Authentication Information

Service APIs need to be authenticated. Two authentication methods, token authentication and AK/SK authentication, are available. You are advised to use AK/SK authentication.

**Step 1** Register with and log in to the management console.

**Step 2** Hover the cursor on the username and select **My Credentials** from the drop-down list.

**Step 3** Click the **Access Keys** tab and then click **Add Access Key**.

**Step 4** Enter the verification code sent to your mail or mobile phone.

**Step 5** Click **OK** to download the AK/SK of the authentication account. The AK/SK data is saved in a local file. Keep the file secure.

**----End**

# 3 Obtaining Moderation SDK

To download the SDK and documents of Content Moderation, see **Content Moderation SDK**.

To obtain the endpoint of Content Moderation, see **Regions and Endpoints**.

# 4 Preparing the Environment

Table 4-1 describes the environment to be prepared for using Moderation SDK.

**Table 4-1** Development environment

| Item | Description |
| --- | --- |
| Operating system (OS) | Windows OS. Windows 7 or later is recommended. |
| JDK | Basic configuration for the Java development environment. The 1.8 version is strongly recommended. |
| Python | Python development environment, compatible with Python 2.6+ and Python 3.*x*. Python 3.*x* is recommended. |
| PHP | Python development environment. PHP 7.*x* is recommended. |
| Node.js | Node.js development environment. Node.js 8.*x* or later is recommended. |

# 5 Using the SDK (Java)

## 5.1 Preparing a Java Development Environment

Moderation Java SDK uses Java SE Development Kit 8 (JDK 8) or later. The following uses JDK 8 (Windows x64) running on Windows 7 as an example. If you have downloaded the JDK and configured the environment, skip this section.

**Step 1** **Download the JDK file.**

**Step 2** After the JDK file is downloaded, install the JDK as prompted. For example, install the JDK to the **C:\Program Files\Java\jdk1.8.0_131** directory on the local PC.

**Step 3** Right-click **Computer**, choose **Properties** > **Advanced System Settings** > **Environment Variables**, and perform the following operations to configure Java environment variables:

1. Create system variable **JAVA_HOME** whose value is the JDK installation path.
2. Add **%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin** to **Path**. Separate multiple values with semicolons (;).
3. Create system variable **CLASSPATH** whose value is **%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar**.

**Step 4** Open the command line interface (CLI) and run **java -version**. If the information shown in **Figure 5-1** is displayed, the configuration is successful.

**Figure 5-1** Java version information

```
C:\>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

**----End**

# 5.2 Installing Eclipse and Importing SDK Projects

The following uses Eclipse as an example to describe how to import the SDK project. Operations of other IDEs are similar to those of Eclipse.

**Step 1** **Download the Eclipse version of the corresponding platform.** For example, **eclipse-jee-mars-R-win32-x86_64.zip**.

**Step 2** Extract and open Eclipse.

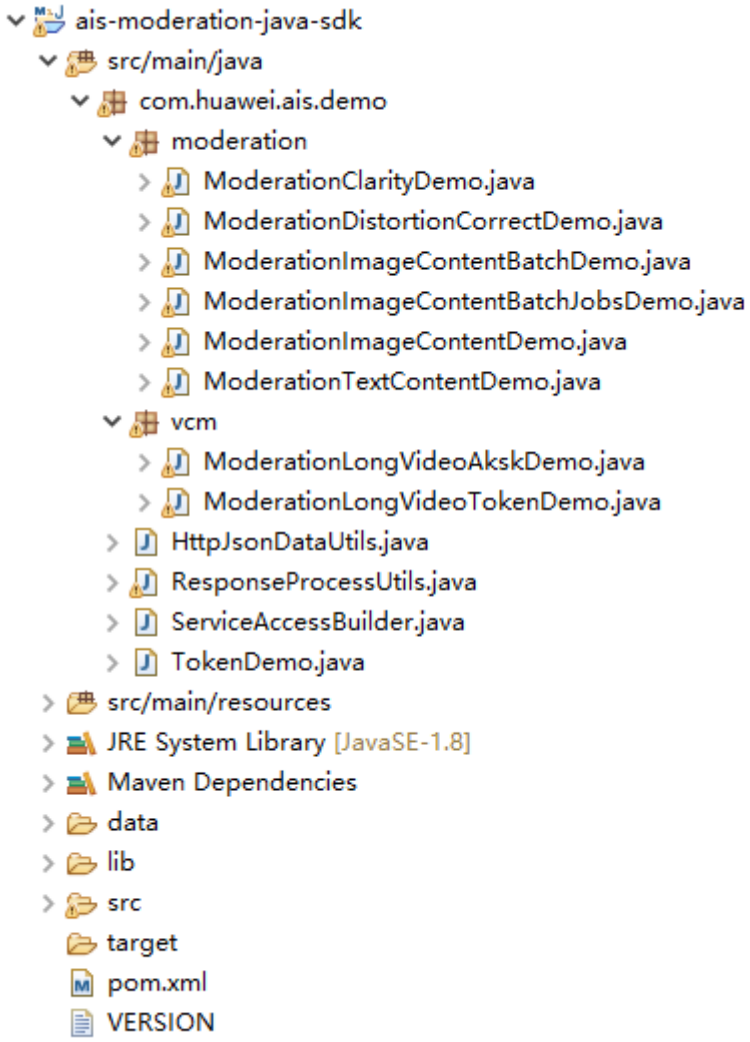
**Step 3** Configure the correct JRE path in **Windows > Preferences > Java > Installed JREs**.

**Step 4** Choose **General > Workspace** from the navigation tree on the left. In the **Text file encoding** area, select **Other**, set the parameter value to **UTF-8**, click **Apply**, and click **OK**.

**Step 5** On the Eclipse menu bar, choose **File > Import**. In the dialog box that is displayed, choose **Maven > Existing Maven Projects**, click **Next**, click **Browse**, and select the local path where **ais-moderation-java-sdk** resides.

**Step 6** Click **Finish** to import the SDK. After the SDK is imported, open the project. **Figure 5-2** shows the project directory.

**Figure 5-2** Project directory




When Maven is used, the JAR package in the **lib** directory of the SDK package is the local JAR package on which the project depends. Other dependencies can be configured in the **pom.xml** file and obtained from the central repository or Huawei open-source image site.

**----End**

# 5.3 Moderation (Text)

## 5.3.1 Demo Project of Text Moderation

This demo project corresponds to the **POST /v1.0/moderation/text** URI. Replace the AK/SK information with the actual AK/SK to run the demo.

## Sample Code

1. Configure the AK and SK pair and region information for accessing a service in the **ModerationTextContentDemo.java** file.

```
// 1. Configure the basic information for accessing Text Moderation and generate a client connection object.
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
       .region("cn-north-4")             // Content Moderation in CN North-Beijing4 (cn-north-4) and CN East-Shanghai1 (cn-east-3)
        .connectionTimeout(5000)          // Timeout limit for connecting to the target URL
        .connectionRequestTimeout(1000)   // Timeout limit for obtaining available connections from the connection pool
        .socketTimeout(20000)             // Timeout limit for obtaining server response data
        .build();
```

2. Enter the text that you want to check in the **ModerationTextContentDemo.java** file, for example, **6666666666**.

```
//
// 2. Construct the parameters required for accessing Text Moderation.
//
String uri = "/v1.0/moderation/text";

JSONObject json = new JSONObject();
json.put("categories", new String[] {"porn","politics","flood"}) //Text for moderation

JSONObject text = new JSONObject();
text.put("text", "6666666666");
text.put("type", "content");

JSONArray items = new JSONArray();
items.add(text);
json.put("items", items);

StringEntity stringEntity = new StringEntity(json.toJSONString(), "utf-8");

// 3. Input the URI and required parameters of Text Moderation.
// Input the parameters in JSON objects and call the service using POST.
HttpResponse response = service.post(uri, stringEntity);

// 4. Check whether the API call is successful. If 200 is returned, the API call succeeds. Otherwise, it fails.
ResponseProcessUtils.processResponseStatus(response);
```

3. Execute the **ModerationTextContentDemo.java** file. If **200** is displayed on the console, the program is successfully executed. The **text moderation result** is displayed on the console. See **Figure 5-3**.

**Figure 5-3** Execution result

```
200
{
    "result": {
        "detail": {
            "flood": "6666666666"
        },
        "suggestion": "block"
    }
}
```

# 5.4 Moderation (Image)

## 5.4.1 Demo Project of Image Moderation

This demo project corresponds to the **POST /v1.0/moderation/image** URI. Replace the AK/SK information with the actual AK/SK to run the demo.

### Sample Code

1. Configure the AK and SK pair and region information for accessing a service in the **ModerationImageContentDemo.java** file.

```
// 1. Configure the basic information for accessing Image Moderation and generate a client connection object.
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
       .region("cn-north-4")            // Content Moderation in CN North-Beijing4 (cn-north-4) and CN East-Shanghai1 (cn-east-3)
        .connectionTimeout(5000)            // Timeout limit for connecting to the target URL
        .connectionRequestTimeout(1000)     // Timeout limit for obtaining available connections from the connection pool
        .socketTimeout(20000)               // Timeout limit for obtaining server response data
        .build();
```

2. Select a local image or use the default image of the sample project, and modify the image file path (**data/moderation-demo-1.jpg**) in the **ModerationImageContentDemo.java** file.

```
public static void main(String[] args) throws IOException {
    ModerationImageContentDemo tool = new ModerationImageContentDemo();
    tool.imageContentCheck("https://sdk-obs-source-save.obs.cn-north-4.myhuaweicloud.com/terrorism.jpg");

    byte[] imageBytes = FileUtils.readFileToByteArray(new File("data/moderation-demo-1.jpg"));
    tool.imageContentCheck(imageBytes);
    }
```

3. Execute the **ModerationImageContentDemo.java** file. If **200** is displayed on the console, the program is successfully executed. The is displayed on the console. See **Figure 5-4**.

**Figure 5-4** Execution result

```
200
{"result":{"detail":{"politics":[],"terrorism":[{"confidence":0.0,"label":"bloody"},{"confidence":0.0,"label":"fire"},{"confidence":0.0,"label":"gun"},{"confidence":0.0,"label":"knife"},{"confidence":0.0,"label":"flag"},{"confidence":0.0,"label":"symbol"},{"confidence":0.0,"label":"dress"},{"confidence":0.0,"label":"war"},{"confidence":0.0,"label":"tiananmen"},{"confidence":1.0,"label":"normal"}]},"suggestion":"pass"}}
```

# 5.5 SDK Projects Using Token Authentication

Two authentication methods, token authentication and AK/SK authentication, are available. This section uses token authentication as an example.

(You can understand porn detection after performing the following operations. You only need to modify the username and password instead of compiling code.)

### Sample Code

1. Set **projectName** in **TokenDemo.java**.

```
/**
 * Access the service using token authentication.
```

```
 */
public class TokenDemo {
private static final String projectName = "cn-north-1";     // Configuration of Content Moderation in
the CN North-Beijing1 (cn-north-1) region.
    private static final String URL_TEMPLATE = ServiceAccessBuilder.getCurrentEndpoint(projectName)
+"/v1.0/moderation/image/batch?job_id=%s";
    private static final long POLLING_INTERVAL = 2000L;
```

2. Open the **TokenDemo.java** file in the **com.huawei.ais.demo** package, and change the values of **username** and **password** in the **main** function to the actual username and password registered with the system. The sample code is as follows:

```
/**
 * Invoke the main entrypoint function.
 */
public static void main(String[] args) throws URISyntaxException, UnsupportedOperationException,
IOException {
    String username = "zhangshan";      // Enter the username.
    String password = "*******";        // Enter the password.
    String domainName = "*******";
    String token = getToken(username, password, domainName, projectName);
    System.out.println(token);

    // Configure the following three timeout limits:
    connectionTimeout = 5000; //Timeout limit for connecting to the target URL
    connectionRequestTimeout = 1000;//Timeout limit for obtaining available connections from the
connection pool
    socketTimeout = 5000;//Timeout limit for obtaining a server response


     //Run Image Clarity Detect.
    //requestModerationClarityBase64(token, "data/moderation-demo-1.jpg");

    //Run Distortion Correction.
    //requestModerationDistortionCorrectBase64(token, "data/moderation-demo-1.jpg");

    //Run Text Moderation.
    //requestModerationTextContentBase64(token, "6666666666");

    //Run Image Moderation.
    //requestModerationImageContentBase64(token, "data/moderation-demo-1.jpg");

     //Run Image Moderation (asynchronous batch task).
    String url1 = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg";
    String url2 = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/antiporn.jpg";
    requestModerationImageContentBatchJobs(token, new String[]{url1,url2});

        //Run Image Moderation (batch task).
    //requestModerationImageContentBatch(token, new String[]{url1,url2});
}
```

3. After modification, run the **TokenDemo.java** file. On the console, you can view the recognition result using token authentication.

# 6 Using Java SDK (Deprecated)

## 6.1 Demo Project of Distortion Correction

This demo project corresponds to the **POST /v1.0/moderation/image/distortion-correct** URI. Replace the AK/SK information with the actual AK/SK to run the demo.

### Sample Code

1. Configure the AK/SK in the **ModerationDistortionCorrectDemo.java** file.

```
// 1. Configure the basic information for accessing Distortion Correction and generate a client connection object.
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
.region("cn-north-1") // Configuration of Content Moderation in the CN North-Beijing1 (cn-north-1) region
        .connectionTimeout(5000)            // Timeout limit for connecting to the target URL
        .connectionRequestTimeout(1000)     // Timeout limit for obtaining available connections from the connection pool
        .socketTimeout(20000)               // Timeout limit for obtaining server response data
        .build();
```

2. Select a local image or use the default image of the sample project, and modify the image file path (**data/moderation-demo-1.jpg**) in the **ModerationDistortionCorrectDemo.java** file.

```
//
// 2. Construct the parameters required for accessing Distortion Correction.
//
String uri = "/v1.0/moderation/image/distortion-correct";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/moderation-demo-1.jpg"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

3. Specify the path for saving the corrected image, for example, **data/moderation-distortion.corrected.jpg**. The sample code example is as follows:

```
// 4. Check whether the API call is successful. If 200 is returned, the API call succeeds. Otherwise, it fails.
if(ResponseProcessUtils.isRespondedOK(response)) {
    ResponseProcessUtils.processResponseWithImage(response, "data/moderation-distortion.corrected.jpg");
} else {
// 5. Process the character stream returned by the service and output the recognition result.
ResponseProcessUtils.processResponseStatus(response);
ResponseProcessUtils.processResponse(response);
}
```

4. Execute the **ModerationDistortionCorrectDemo.java** file. If **200** is displayed on the console, the program is successfully executed. The **recognition result** is displayed on the console, as shown in **Figure 6-1**.

   The returned image is saved in the file specified in **3**, as shown in **Figure 6-2**.

   **Figure 6-1** Execution result

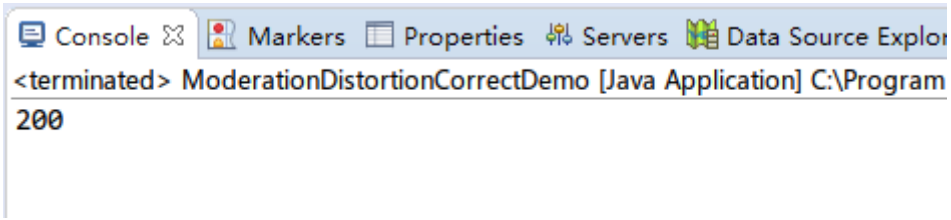
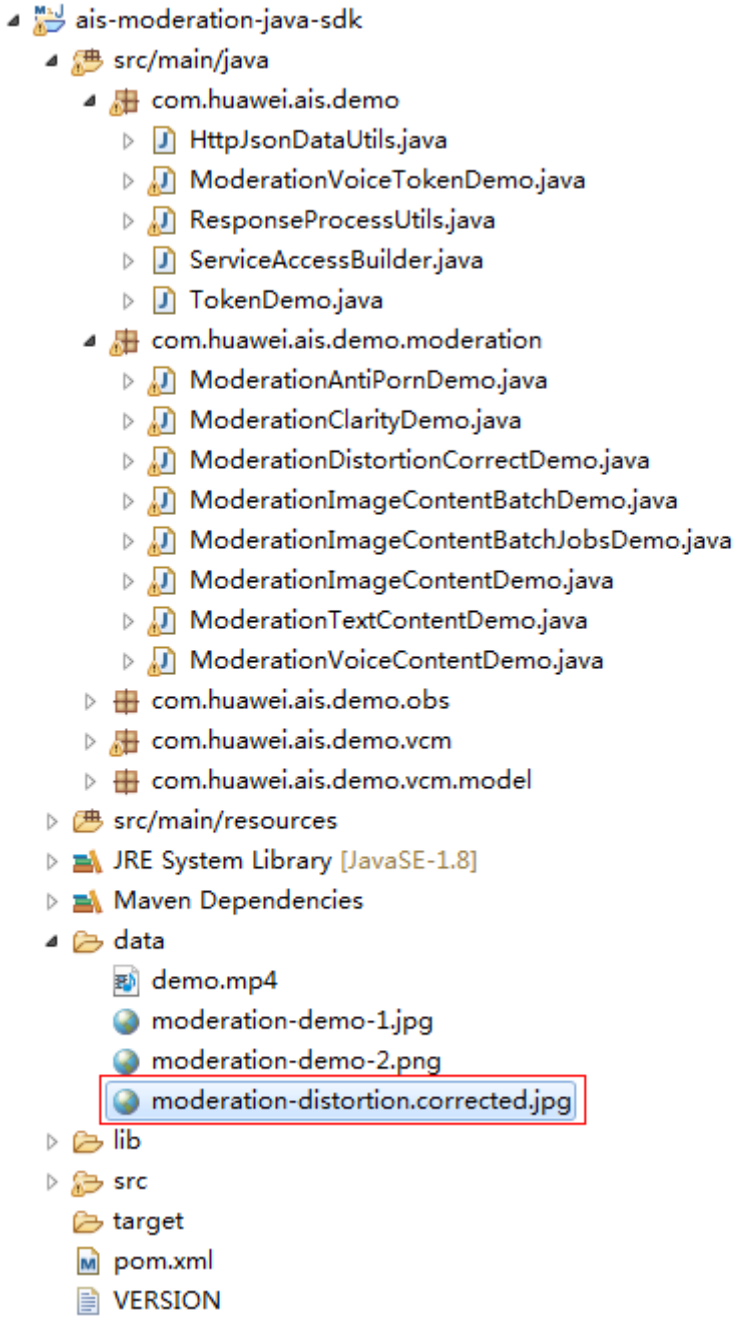


**Figure 6-2** Returned image file




# 6.2 Demo Project of Image Clarity Detect

This demo project corresponds to the **POST /v1.0/moderation/image/clarity-detect** URI. Replace the AK/SK information with the actual AK/SK to run the demo.

## Sample Code

1. Configure the AK/SK in the **ModerationClarityDemo.java** file.

```
// 1. Configure the basic information for accessing Image Clarity Detect and generate a client connection object.
```

```
AisAccess service = ServiceAccessBuilder.builder()
        .ak("#######")                    // your ak
        .sk("#######")                    // your sk
.region("cn-north-1") // Configuration of Content Moderation in the CN North-Beijing1 (cn-north-1) region
        .connectionTimeout(5000)            // Timeout limit for connecting to the target URL
        .connectionRequestTimeout(1000)     // Timeout limit for obtaining available connections from the connection pool
        .socketTimeout(20000)               // Timeout limit for obtaining server response data
        .build();
```

2. Select a local image or use the default image of the sample project, and modify the image file path (**data/moderation-demo-1.jpg**) in the **ModerationClarityDemo.java** file.

```
//
// 2. Construct the parameters required for accessing Image Clarity Detect.
//
String uri = "/v1.0/moderation/image/clarity-detect";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/moderation-demo-1.jpg"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

3. Execute the **ModerationClarityDemo.java** file. If **200** is displayed on the console, the program is successfully executed. The **recognition result** is displayed on the console, as shown in **Figure 6-3**.

**Figure 6-3** Execution result

```
200
{
  "result": {
    "category": "blur",
    "detail": [
      {
        "label": "clarity",
        "confidence": 0.0
      },
      {
        "label": "blur",
        "confidence": 1.0
      }
    ]
  }
}
```

# 7 Using the Python SDK

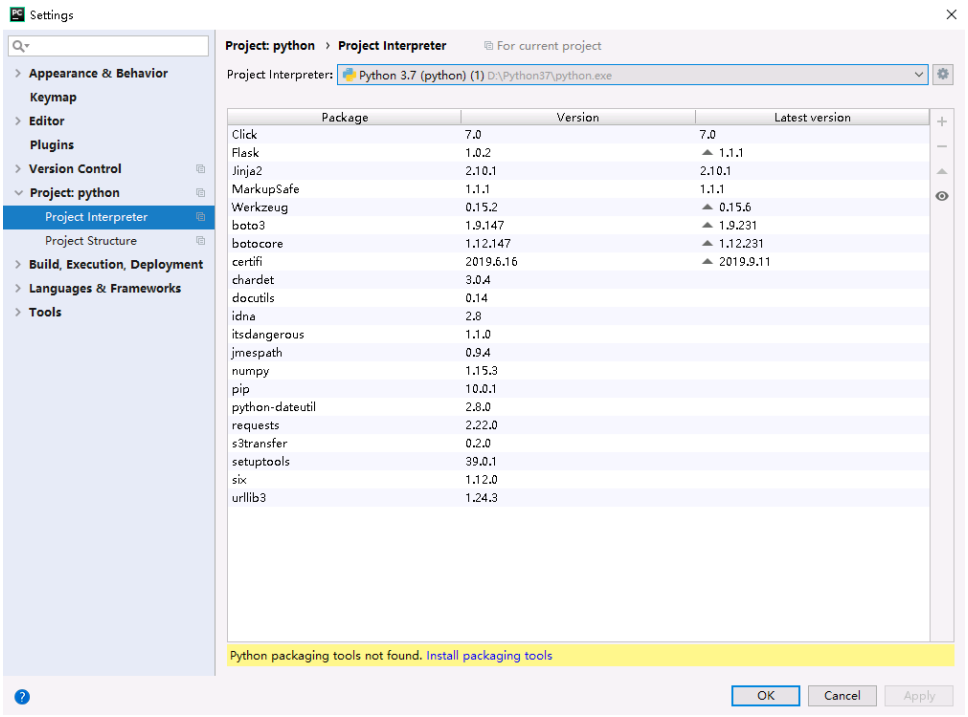
## 7.1 Preparing a Python Development Environment

To use the Python SDK of Content Moderation, you need to configure the Python development environment.

1. Download Python of a proper version from **Python's official website** and install it. Python 2.6+ and Python 3.*x* are supported. Python 3.*x* is recommended. The following uses Python 3.7 as an example.
2. Download the latest version of PyCharm from **PyCharm's official website**.
3. Start the PyCharm development tool and choose **File > Settings > Project Interpreter** to configure the Python environment.
4. Select the Python installation path, as shown in **Figure 7-1**. After selecting the target Python, click **Apply** at the bottom of the page to complete the configuration.

**Figure 7-1** Configuring the python environment using PyCharm




# 7.2 Importing a Python SDK into a Python Environment

1. On the **HUAWEI CLOUD SDKs** page, download and decompress the Content Moderation Python SDK.

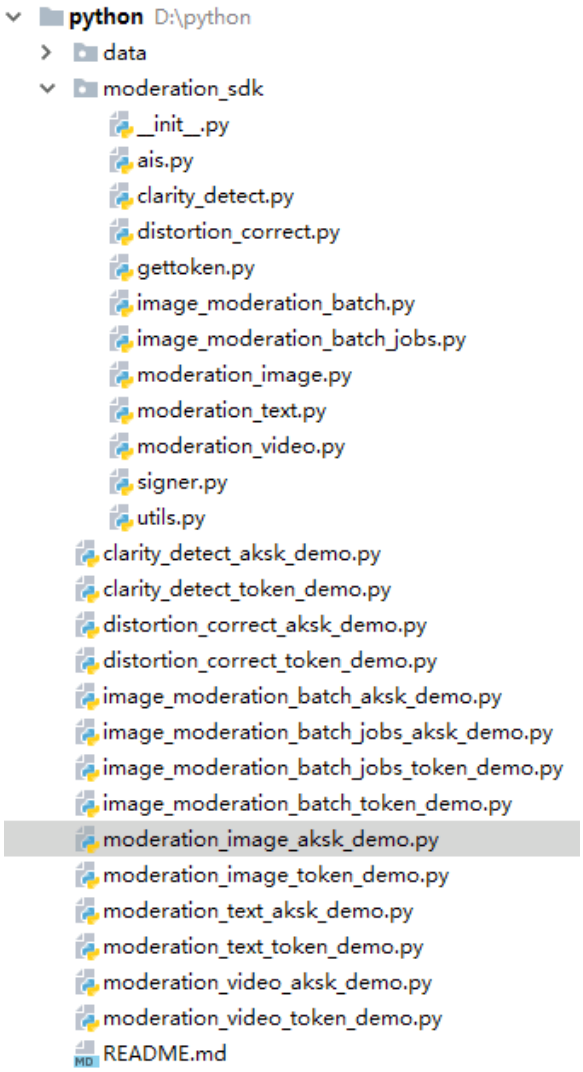
   The structure and content of the decompressed package are described as follows:

```
├──python
│   ├── data                          # Sample image
│   ├── moderation_sdk                # Content Moderation SDK code
│   │   ├── ais.py                    # Constant configuration of Content Moderation
│   │   ├── moderation_image.py       # Image Moderation SDK code
│   │   ├── signer.py
│   ├── moderation_image_aksk_demo.py  # Image Moderation AK/SK demo
│   ├── moderation_image_token_demo.py  # Image Moderation token demo
```

2. On the PyCharm page, choose **File > Open File or Project**, select the path for storing the decompressed SDK package, and import the SDK package. See **Figure 7-2**.

**Figure 7-2** Project directory after importing the SDK




# 7.3 SDK Calling Example

There are two service authentication modes: This section uses Image Moderation as an example to describe how to use **AK/SK-based** and **token-based** authentication. The corresponding URI is **POST /v1.0/moderation/image**.

## Using SDK in AK/SK-based Authentication Mode

1. Set **app_key** and **app_secret** in the **moderation_image_aksk_demo.py** file. The sample code is as follows:

```
if __name__ == '__main__':
    # Services currently support North China-Beijing(cn-north-4)
    init_global_env('cn-north-4')
    #
    # access moderation image,post data by ak,sk
    #
    app_key = '**************'
    app_secret = '************'
```

2. Modify the local path or URL path of the image in the **moderation_image_aksk_demo.py** file. Image Moderation supports two calling methods: file calling and URL calling.
   - To call the file, modify the local path of the target image. Specifically, replace **data/moderation-terrorism.jpg** in **encode_to_base64** with the path of the target image. The sample code is as follows:
     ```
     # call interface use the local file
     result = moderation_image_aksk(app_key, app_secret, encode_to_base64('data/moderation-terrorism.jpg'), '', ['politics','terrorism'],'')
     print(result)
     ```
   - To call the URL, modify the URL of the target image. Specifically, replace the image URL of **demo_data_url** with the URL of the target image. The sample code is as follows:
     ```
     demo_data_url = 'https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg'
     # call interface use the url
     result = moderation_image_aksk(app_key, app_secret, "", demo_data_url, ['politics','terrorism'], '')
     print(result)
     ```
3. Execute the **moderation_image_aksk_demo.py**. If the moderation result is displayed on the console, the execution is successful.
   ```
   {"result":{"detail":{"porn":[{"confidence":1.0,"label":"normal"},{"confidence":0.0,"label":"porn"},{"confidence":0.0,"label":"sexy"}]},"suggestion":"pass","category_suggestions":{"porn":"pass"}}}
   {"result":{"detail":{"porn":[{"confidence":1.0,"label":"normal"},{"confidence":0.0,"label":"porn"},{"confidence":0.0,"label":"sexy"}]},"suggestion":"pass","category_suggestions":{"porn":"pass"}}}

   Process finished with exit code 0
   ```

## Using SDK in Token-based Authentication Mode

1. In the **moderation_image_token_demo.py** file, set the registered username and password. The username of a non-IAM login user is the same as the account name of an IAM login user. The sample code is as follows:
   ```
   if __name__ == '__main__':

       # Services currently support North China-Beijing(cn-north-4)
       init_global_env('cn-north-4')
       #
       # access moderation image,post data by token
       #
       user_name = '******'
       password = '******'
       account_name = '******'  # the same as user_name in commonly use

       token = get_token(user_name, password, account_name)
   ```
2. Modify the local path or URL path of the image in the **moderation_image_token_demo.py** file. Image Moderation supports two calling methods: file calling and URL calling.
   - To call the file, modify the local path of the target image. Specifically, replace **data/moderation-terrorism.jpg** in **encode_to_base64** with the path of the target image. The sample code is as follows:
     ```
     # call interface use the local file
     result = moderation_image(token, encode_to_base64('data/moderation-terrorism.jpg'), '', ['porn'], '')
     print(result)
     ```
   - To call the URL, modify the URL of the target image. Specifically, replace the image URL of **demo_data_url** with the URL of the target image. The sample code is as follows:
     ```
     demo_data_url = 'https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg'
     # call interface use the url (token, image, url, threshold=0.95, scene=None)
     ```

```
result = moderation_image(token, "", demo_data_url, ['porn'], '')
print(result)
```

3. Execute the **moderation_image_token_demo.py** file. If the moderation result is displayed on the console, the execution is successful.

```
{"result":{"detail":{"porn":[{"confidence":1.0,"label":"normal"},{"confidence":0.0,"label":"porn"},
{"confidence":0.0,"label":"sexy"}]},"suggestion":"pass","category_suggestions":{"porn":"pass"}}}
{"result":{"detail":{"porn":[{"confidence":1.0,"label":"normal"},{"confidence":0.0,"label":"porn"},
{"confidence":0.0,"label":"sexy"}]},"suggestion":"pass","category_suggestions":{"porn":"pass"}}}

Process finished with exit code 0
```

# 8 Using the PHP SDK

## 8.1 Preparing a PHP Development Environment

To use the PHP SDK of Content Moderation, you need to configure the PHP development environment.

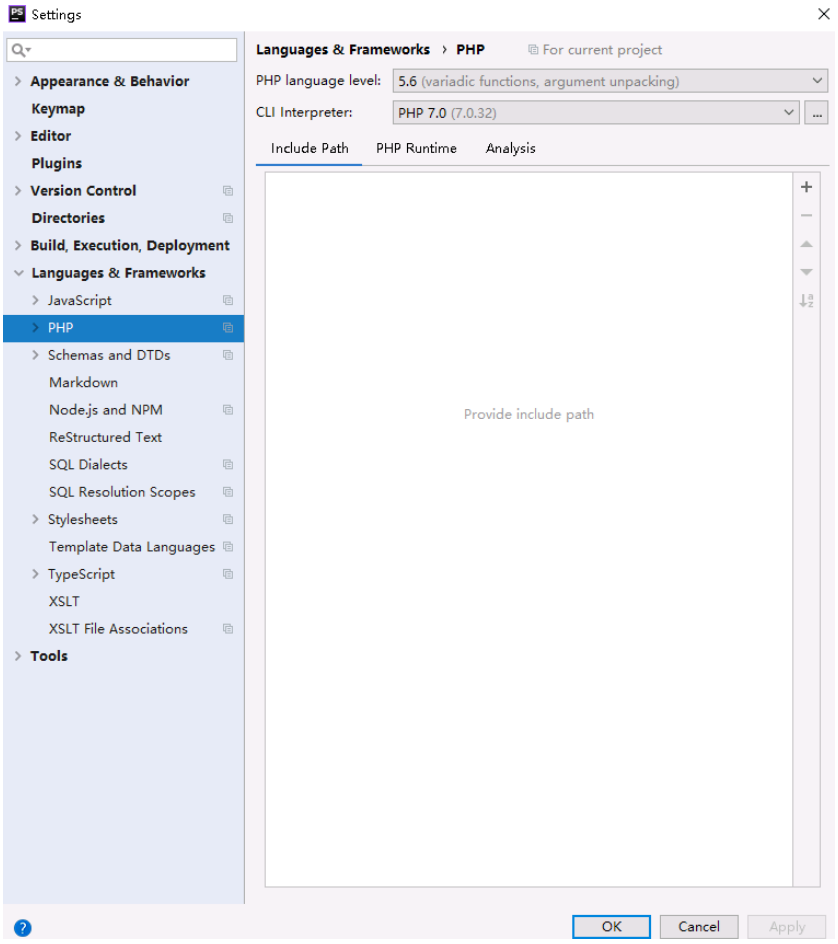
1. Download PHP of a proper version from **PHP's official website** and install it. The php.7.*x* version is recommended for Content Moderation SDK. The following uses the php 7.0.32 version as an example.

   NOTE

   - The PHP installation directory cannot contain Chinese characters or special characters.
   - In the Windows operating system, if you run the **php –v** command, the command output indicates that the specified module cannot be found, check whether the **extension_dir** and **include_path** parameters in the **php.ini** file are correctly set.
2. Download the latest version of PhoStorm from **PhoStorm's official website** and install it.
3. Start the PhpStorm development tool and choose **File > Settings > Languages & Frameworks > PHP** to configure the PHP environment.
4. Select the PHP path on the top of the page, as shown in **Figure 8-1**. Then click **Apply**.

**Figure 8-1** Configuring the PHP environment using PhpStorm




# 8.2 Importing a PHP SDK into a PHP Environment

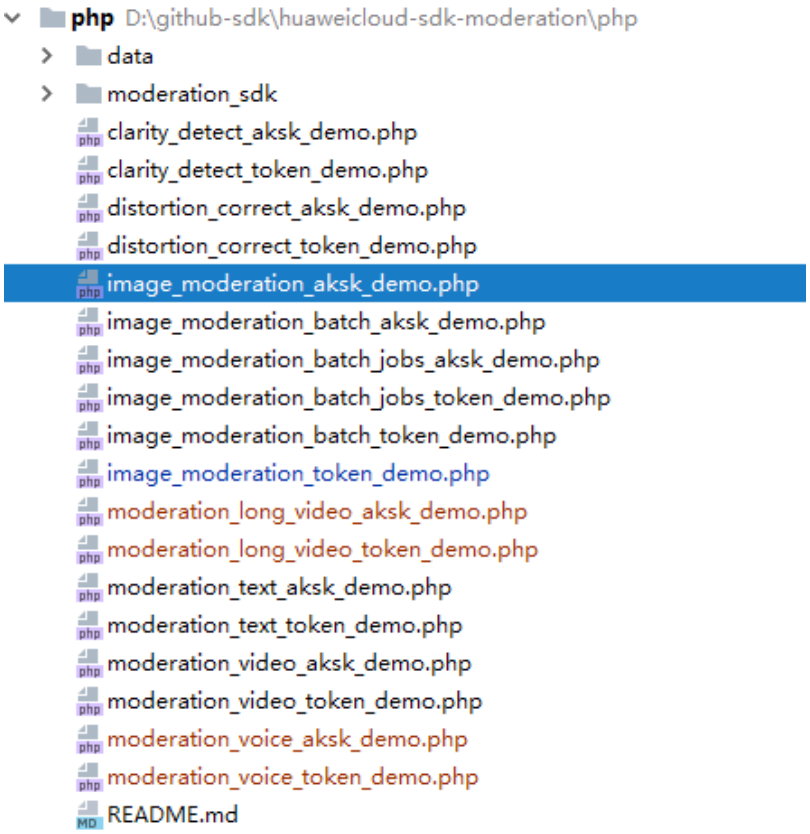
1. On the **Content Moderation SDKs** page, download and decompress the Content Moderation PHP SDK. The structure and content of the decompressed package are described as follows:

```
├── php
│   ├── data                          # Sample image
│   ├── moderation_sdk                # Content Moderation SDK code
│   │   ├── ais.py                    # Constant configuration of Content Moderation
│   │   ├── moderation_image.py       # Image Moderation SDK code
│   │   ├── signer.py
│   ├── image_moderation_token_demo.php    # Image Moderation AK/SK demo
│   ├── image_moderation_aksk_demo.php     # Image Moderation token demo
```

2. On the PhpStorm page, choose **File > Open File or Project** select the path for storing the decompressed SDK package, and import the SDK package. See **Figure 8-2**.

**Figure 8-2** Project directory after importing the SDK




# 8.3 SDK Calling Example

There are two service authentication modes: This section uses Image Moderation as an example to describe how to use **AK/SK-based** and **token-based** authentication. The corresponding URI is **POST /v1.0/moderation/image**.

## Using SDK in AK/SK-based Authentication Mode

1. Set **app_key** and **app_secret** in the **image_moderation_aksk_demo.php** file. The sample code is as follows:

```
//CN North-Beijing1 (cn-north-4) is supported.
init_region($region = 'cn-north-4');

$app_key = "**************";
$app_secret = "**************";
```

2. Modify **data_url** or **filepath** in the **image_moderation_aksk_demo.php** file. Image Moderation supports two calling methods: file calling and URL calling.
   - To call the file, modify the local path of the target image. Specifically, replace **./data/moderation-terrorism.jpg** in **filepath** with the path of the target image. The sample code is as follows:

```
$filepath = "./data/moderation-terrorism.jpg";
$data = file_to_base64($filepath);

$result = image_content_aksk($app_key, $app_secret, $data, "", array("politics"), 0);
echo $result;
```

- To call the URL, modify the URL of the target image. Specifically, replace the image URL of **data_url** with the URL of the target image. The sample code is as follows:

```
$data_url = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg";

$result = image_content_aksk($app_key, $app_secret, "", $data_url, array("politics"), 0);
echo $result;
```

3. Execute the **image_moderation_aksk_demo.php** file. If the moderation result is displayed on the console, the execution is successful.

```
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
Process finished with exit code 0
```

## Using SDK in Token-based Authentication Mode

1. In the **image_moderation_token_demo.php** file, set the registered username and password. The username of a non-IAM login user is the same as the domain name of an IAM login user. The sample code is as follows:

```
//CN North-Beijing1 (cn-north-4) is supported.
init_region($region = 'cn-north-4');

$username = "********"; // Configure the username.
$password = "********";      // Configure the password.
$domainName = "**********"; // Configure the domain name.

$token = get_token($username, $password, $domainName);
```

2. Modify **data_url** or **filepath** of the image in the **image_moderation_token_demo.php** file. Image Moderation supports two calling methods: file calling and URL calling.

   - To call the file, modify the local path of the target image. Specifically, replace **./data/moderation-terrorism.jpg** in **filepath** with the path of the target image. The sample code is as follows:

```
$filepath = "./data/moderation-terrorism.jpg";
$data = file_to_base64($filepath);

$result = image_content($token, $data, "", array("politics"), 0);
echo $result;
echo "\n";
```

   - To call the URL, modify the URL of the target image. Specifically, replace the image URL of **data_url** with the URL of the target image. The sample code is as follows:

```
$data_url = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg";

$result = image_content($token, "", $data_url, array("politics"), 0);
echo $result;
```

3. Execute the **image_moderation_token_demo.php** file. If the moderation result is displayed on the console, the execution is successful.

```
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
Process finished with exit code 0
```

# 9 Using Node.js SDK

## 9.1 Preparing a Node.js Development Environment

The JavaScript SDK of Content Moderation depends on the Node.js environment.

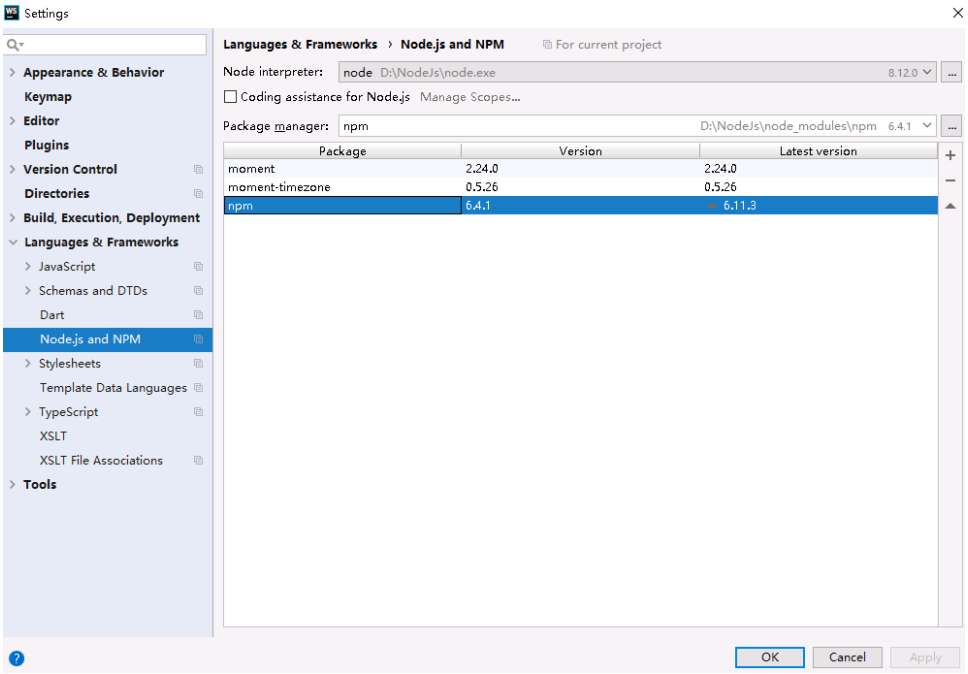
1. Download the recommended version from the **Node.js's official website** and install it. Node.js 8.*x* or later is recommended. The following uses Node.js 8.12.0 as an example.
2. Download the latest version of WebStorm from **WebStorm's official website**.
3. On the WebStorm page, choose **File > Settings > Languages & Frameworks > Node.js and NPM** to configure the Node.js environment.
4. Select the **Node.js and NPM** installation path, as shown in **Figure 9-1**. After selecting the target **Node.js and NPM**, click **Apply** at the bottom of the page to complete the configuration.

   NOTE

   The JavaScript SDK of Content Moderation needs to depend on the moment-timezone and moment modules. You can run the **npm install** command to install the SDK, or click **+** in **Figure 9-1** to install the SDK using a tool.

**Figure 9-1** Configure the Node.js environment




# 9.2 Importing a Node.js SDK into a Node.js Environment

1. On the **HUAWEI CLOUD SDKs** page, download and decompress the Content Moderation Node.js SDK. The structure and content of the decompressed package are described as follows:

```
├── javascript
│   ├── data                          # Sample image
│   ├── moderation_sdk                # Content Moderation SDK code
│   │   ├── ais.py                    # Constant configuration of Content Moderation
│   │   ├── image_moderation.js       # Image Moderation SDK code
│   │   └── signer.js
│   ├── moderation_image_aksk_demo.py     # Image Moderation AK/SK demo
│   ├── moderation_image_token_demo.py    # Image Moderation token demo
│   └── README.md                     # Description and precautions of Content Moderation SDK
```

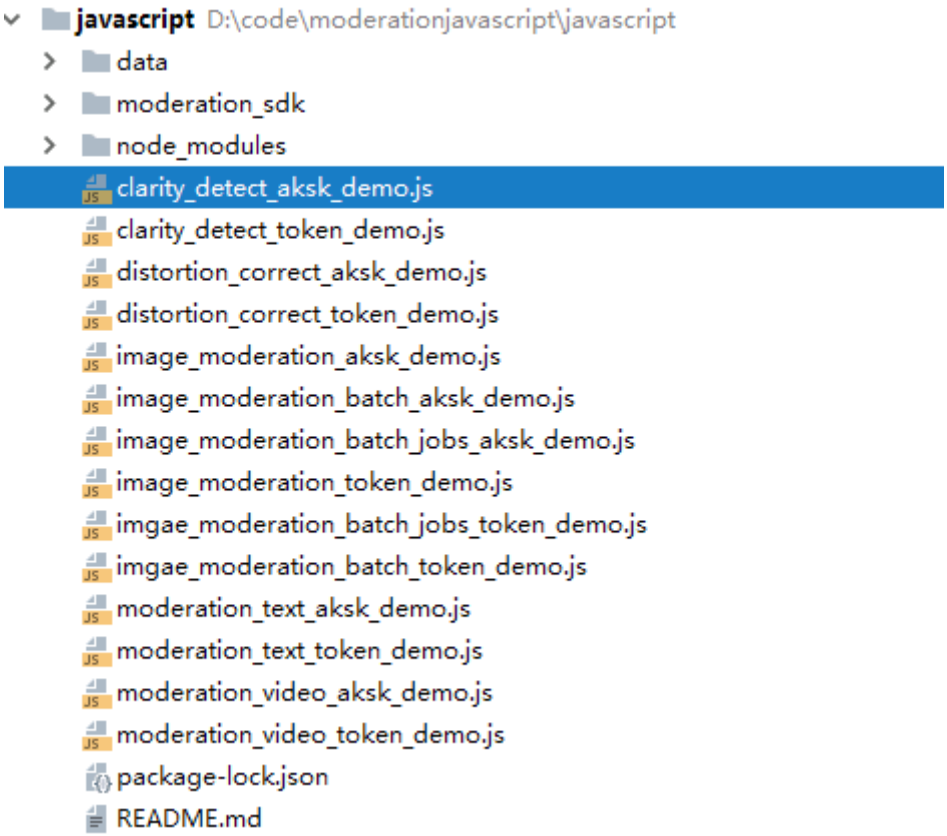
2. On the WebStorm page, choose **File > Open File or Project**, select the path for storing the decompressed SDK package, and import the SDK package. See **Figure 9-2**.

NOTE

**node_modules** is the installation path of the JavaScrip SDK dependency package of Content Moderation. It depends on the **moment-timezone** and **moment** modules.

**Figure 9-2** Project directory after importing the SDK




# 9.3 SDK Calling Example

There are two service authentication modes: This section uses Image Moderation as an example to describe how to use **AK/SK-based** and **token-based** authentication. The corresponding URI is **POST /v1.0/moderation/image**.

## Using SDK in AK/SK-based Authentication Mode

1. Set **app_key** and **app_secret** in the **image_moderation_aksk_demo.js** file. The sample code is as follows:

```
// CN North-Beijing1 (cn-north-4) is supported.
utils.initRegion("cn-north-4");
var app_key = "*************";
var app_secret = "*************";
```

2. Modify **data_url** or **filepath** of the image in the **image_moderation_aksk_demo.js** file. Image Moderation supports two calling methods: file calling and URL calling.
   - To call the file, modify the local path of the target image. Specifically, replace **./data/moderation-terrorism.jpg** in **filepath** with the path of the target image. The sample code is as follows:

```
var filepath = "./data/moderation-terrorism.jpg";
var data = utils.changeFileToBase64(filepath);

content.image_content_aksk(app_key, app_secret, data, "", ["politics"], "", function (result) {
    console.log(result);
});
```

- To call the URL, modify the URL of the target image. Specifically, replace the image URL of **demo_data_url** with the URL of the target image. The sample code is as follows:

```
// The OBS link must be consistent with the region. The OBS resources cannot be shared among different regions.
demo_data_url = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg";

content.image_content_aksk(app_key, app_secret, "", demo_data_url, ["politics"], "", function (result) {
    console.log(result);
});
```

3. Execute the **image_moderation_aksk_demo.js** file. If the moderation result is displayed on the console, the execution is successful.

```
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}

Process finished with exit code 0
```

## Using SDK in Token-based Authentication Mode

1. In the **image_moderation_token_demo.js** file, set the registered username and password. The username of a non-IAM login user is the same as the domain name of an IAM login user. The sample code is as follows:

```
// CN North-Beijing1 (cn-north-4) is supported.
utils.initRegion("cn-north-4");
var username = "********"; // Configure the username.
var domain_name = "********"; // Configure the domain name.
var password = "********"; // Configure the password.
```

2. Modify **demo_data_url** or **filepath** of the image in the **image_moderation_token_demo.js** file. Image Moderation supports two calling methods: file calling and URL calling.

   - To call the file, modify the local path of the target image. Specifically, replace **./data/moderation-terrorism.jpg** in **filepath** with the path of the target image. The sample code is as follows:

```
var filepath = "./data/moderation-terrorism.jpg";
var data = utils.changeFileToBase64(filepath);

token.getToken(username, domain_name, password, function (token) {
    content.image_content(token, data, "", ["politics"], "", function (result) {
        console.log(result);
    });
});
```

   - To call the URL, modify the URL of the target image. Specifically, replace the image URL of **demo_data_url** with the URL of the target image. The sample code is as follows:

```
// The OBS link must be consistent with the region. The OBS resources cannot be shared among different regions.
demo_data_url = "https://ais-sample-data.obs.cn-north-1.myhuaweicloud.com/terrorism.jpg";

token.getToken(username, domain_name, password, function (token) {
    content.image_content(token, "", demo_data_url, ["politics"], "", function (result) {
        console.log(result);
    })
});
```

3. Execute the **image_moderation_token_demo.js** file. If the moderation result is displayed on the console, the execution is successful.

```
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}
{"result":{"detail":{"politics":[]},"suggestion":"pass","category_suggestions":{"politics":"pass"}}}

Process finished with exit code 0
```

# 10 Change History

| Release Date | What's New |
| --- | --- |
| 2019-09-25 | This is the eighth official release.<br>Added the following content:<br>• **Using the Python SDK**<br>• **Using the PHP SDK**<br>• **Using the Node.js SDK** |
| 2019-05-15 | This is the seventh official release.<br>Modified the following content:<br>• **Obtaining Authentication Information**<br>• **Using the SDK (Java)** |
| 2019-01-31 | This is the sixth official release.<br>Modified the following content:<br>• Changed the endpoint of Content Moderation to **moderation.cn-north-1.myhuaweicloud.com**. |
| 2018-12-14 | This is the fifth official release.<br>Added the following content:<br>• Introduction to Moderation SDK<br>• **Preparing the Environment** |
| 2018-10-23 | This is the fourth official release. |
| 2018-04-20 | This is the third official release.<br>Added the following content:<br>• **Demo Project of Text Moderation**<br>• **Demo Project of Image Moderation** |

| Release Date | What's New |
| --- | --- |
| 2018-03-30 | This is the second official release.<br>Added the following content:<br>• **Demo Project of Image Clarity Detect**<br>• **Demo Project of Distortion Correction** |
| 2018-01-03 | This is the first official release. |