Optical Character Recognition

Getting Started

Issue 01

Date 2021-04-02

HUAWEI TECHNOLOGIES CO., LTD.

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 Introduction to OCR

Optical Character Recognition (OCR) recognizes and converts characters and digits in an image into editable text.

OCR provides the text recognition capability through open application programming interfaces (APIs). To call APIs and process the recognition result returned in JSON format, you need to have basic knowledge of programming.

Use either of the following methods to call OCR APIs:

- **Visualization tool (such as curl or Postman)**

  Apply to developers who are familiar with coding, HTTP requests, and API calling. For details, see **Using Postman to Call OCR**.

- **Software development kit (SDK)**

  Apply to developers who are familiar with coding and allow you to use SDKs for Java, Python, iOS, Android, and Node.js for quick integration. For details, see **Using SDK to Call OCR**.

2 Using Postman to Call OCR

Use Postman as an example to help you quickly experience and understand how to call an OCR API without coding, for example, ID Card OCR.

To call an OCR API using Postman, do the following:

Step 1: Subscribing to a Service

Step 2: Configuring the Environment

Step 3: Using a Token for Authentication

Step 4: Calling the Service

Preparations

You have registered an account with HUAWEI CLOUD and completed identity authentication. Your account cannot be in arrears or frozen.

Step 1: Subscribing to a Service

1. Log in to the HUAWEI CLOUD OCR console.

   The service is deployed in the **CN North-Beijing4** region by default. Select a region based on service requirements. For details about the region where each service is deployed, see **Regions and Endpoints**.

2. Select and subscribe to your desired API.

   For this example, subscribe to the ID Card OCR API.

Step 2: Configuring the Environment

1. Download and install Postman 7.24.0.
2. Download the Postman configuration file for OCR.

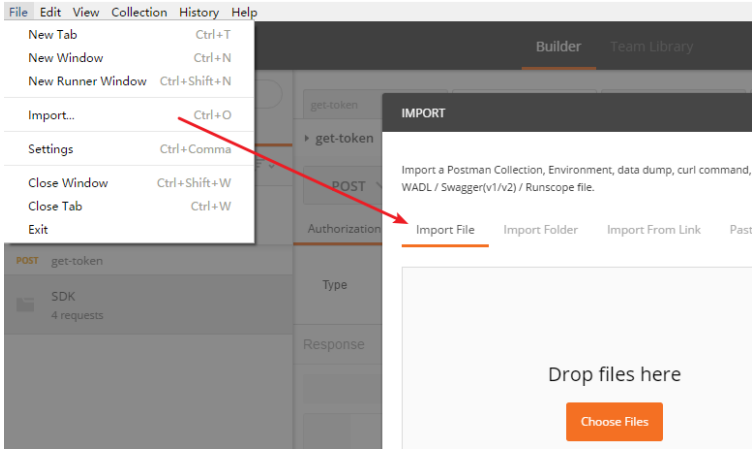
   Download link: **SDK.postman_collection_v2.json**

3. Import the configuration file.

   a. Open and log in to Postman.

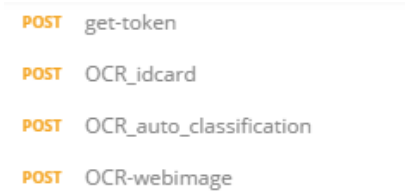
   b. Choose **File** > **Import** > **Import File** > **Choose Files** to import the configuration file.

Figure 2-1 Importing the configuration file



After the configuration file is imported, it is displayed in the left navigation pane.

Figure 2-2 Modifying the configuration file



NOTE

This configuration file uses the **CN North-Beijing4** region by default. If you select another region when subscribing to the service, replace all **cn-north-4** fields in the configuration file with the actual region when performing steps 3 to 4. For details about the region where each service is deployed, see **Regions and Endpoints**.

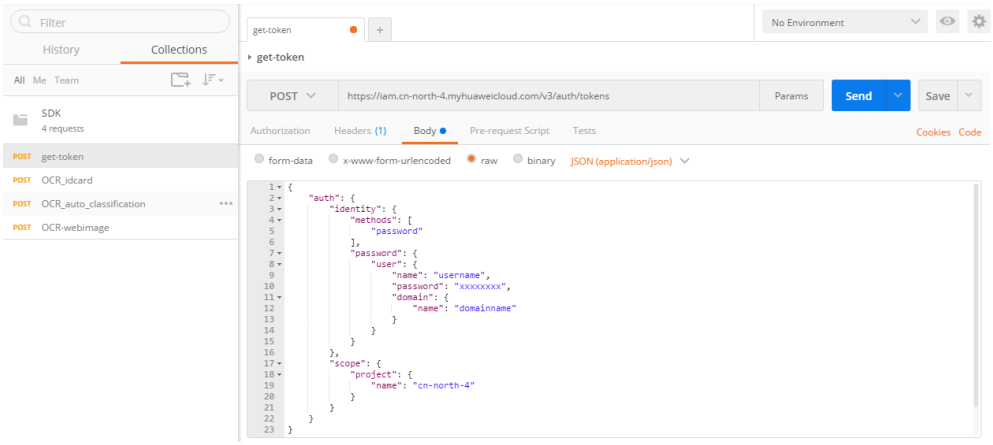
Step 3: Using a Token for Authentication

A token is a user's access credential, which includes user identities and permissions. When you call an API to access a cloud service, a token is required for identity authentication.

1. In the navigation pane of Postman, click the **get-token** configuration file.
2. Click **Body**. Enter the username, password, and domain name.

   Log in to the **My Credential** page, copy the IAM username and account name to **username** and **domain name** respectively, and enter the password.

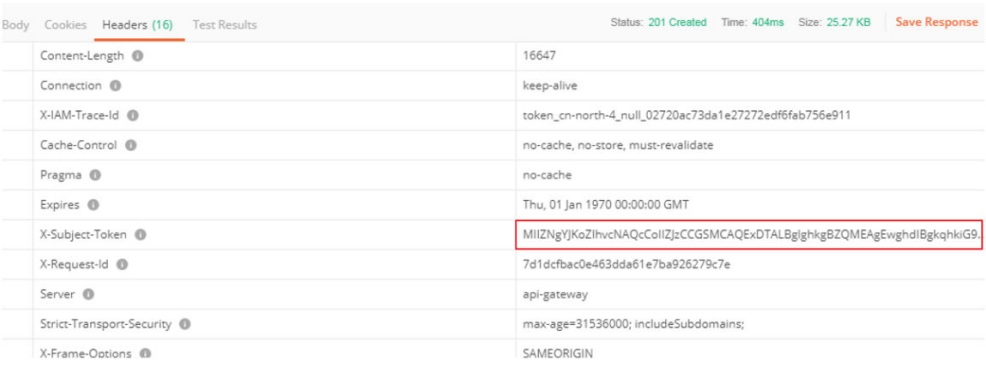
Figure 2-3 Token-based authentication



3. Click **Send** to send a request, and obtain and copy the token.

   Obtain the value of **X-Subject-Token** in **Headers**, which is the token. The token is valid for 24 hours.

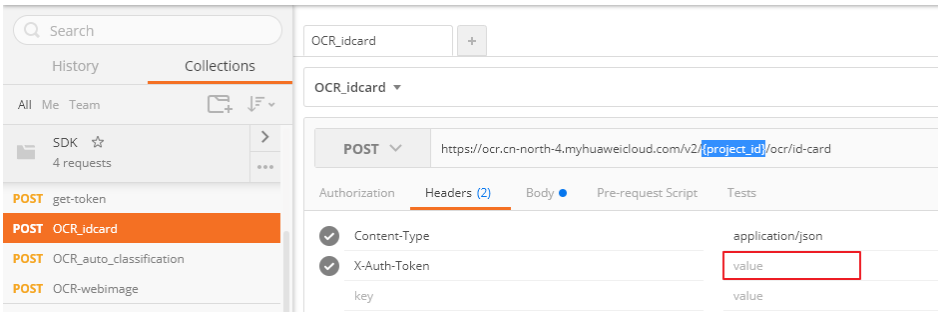
Figure 2-4 Obtaining a token



Step 4: Calling the Service

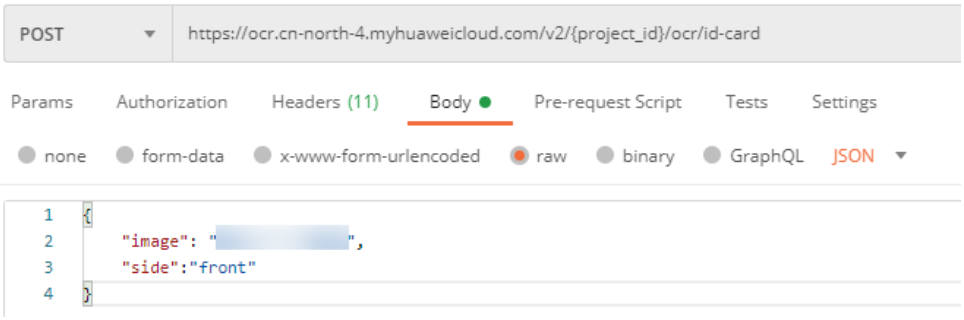
1. In the navigation pane of Postman, click the **OCR_idcard** configuration file.
2. Log in to the **My Credential** page, obtain the project ID of the **CN North-Beijing4** region, and replace *{project_id}* in the URL of the configuration file with the obtained project ID.
3. Click **Headers** and copy the token value to **X-Auth-Token**.

Figure 2-5 Modifying the configuration file



4. Click **Body** and enter the Base64 code of the image to **image**.
   - In the example, the value of **side** is **front**, indicating that the front of the ID card is recognized. For details about the API description, see **API Description**.
   - For details about how to obtain the Base64 code of an image, see **How Do I Obtain the Base64 Code of an Image?**

   Figure 2-6 Modifying the configuration file



5. Click **Send** to send the request and obtain the API calling result.

```
{
   "result": {
      "name": "xx",
      "sex": "Gender recognized from the image",
      "ethnicity": "Ethnicity recognized from the image",
      "birth": "1990-xx-xx",
      "address": "Address recognized from the image",
      "number": "3892011990012xxxxx"
   }
}
```


3 Using SDK to Call OCR

OCR SDKs encapsulate OCR RESTful APIs to simplify development. Currently, OCR SDKs for Java, Python, iOS, Android, Node.js, PHP, C++, and C# are available.

Use the SDK for Java as an example to describe how to call an OCR API, for example, ID Card OCR. You can use the SDK by only calling API functions.

To call an OCR API using an SDK, do the following:

Step 1: Subscribing to a Service

Step 2: Configuring the Environment

Step 3: Modifying the Configuration

Step 4: Calling the Service

Preparations

You have registered an account with HUAWEI CLOUD and completed identity authentication. Your account cannot be in arrears or frozen.

Step 1: Subscribing to a Service

1. Log in to the HUAWEI CLOUD OCR console.

   The service is deployed in the **CN North-Beijing4** region by default. Select a region based on service requirements. For details about the region where each service is deployed, see **Regions and Endpoints**.

2. Select and subscribe to your desired API.

   For this example, subscribe to the ID Card OCR API.

Step 2: Configuring the Environment

1. Download the OCR SDK for Java.

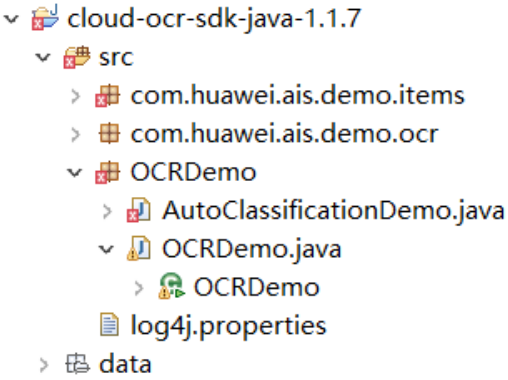
   Download link: **https://developer.huaweicloud.com/sdk?OCR**

2. Prepare a Java development environment.
   - Download a JDK from the **Oracle official website** and install it.
   - Download Eclipse IDE for Java Developers of the latest version from the **Eclipse official website** and install it.

For details about how to configure the environment, see **Preparing a Java Development Environment**.

3. Import the OCR SDK for Java project.
   a. Open Eclipse, and configure the correct JRE path in **Window** > **Preferences** > **Java** > **Installed JREs**.
   b. Right-click on the **Package Explorer** page, click **Import**, choose **General** > **Existing Projects into Workspace**, and select the path of the OCR SDK demo package.
   c. Click **Finish** to import the demo.

Figure 3-1 Demo of OCR SDK for Java



NOTE

The OCR SDK calls the service from the **CN North-Beijing4** region by default. If you select another region when subscribing to the service, replace all **cn-north-4** fields in the configuration file with the actual region when performing step 3. For details about the region where each service is deployed, see **Regions and Endpoints**.

Step 3: Modifying the Configuration

AK/SK- or token-based authentication can be used for the demo of OCR SDK for Java. This example uses AK/SK-based authentication.

1. Obtain an AK/SK.

   The AK/SK is the access key. To obtain the AK/SK, log in to the **My Credential** page and choose **Access Keys** > **Add Access Key**.

2. For AK/SK-based authentication, configure the AK/SK in the OCR SDK for Java.

   Change the values of **AK** and **SK** of the function in the **OCRDemo.java** file of the demo project to the obtained AK/SK.

Figure 3-2 Configuring the AK/SK

```
public static void AKSKDemo() throws URISyntaxException, UnsupportedOperationException, IOException{
    /*
     * AK/SK demo code
     * */
    String AK="xxx"; //AK from authentication
    String SK="xxx"; //SK from authentication
    String regionName="cn-north-4"; //region name of the service
    String httpUri = "/v1.0/ocr/id-card";
    String imgPath = "./data/id-card-demo.jpg"; //File path or URL of the image to be recognized.
```


3. Modify the **OCRDemo.java** configuration file in the OCR SDK for Java.

Figure 3-3 Modifying the OCRDemo.java file

```
public static void AKSKDemo() throws URISyntaxException, UnsupportedOperationException, IOException{
    /*
     * AK/SK demo code
     * */
    String AK="xxx"; //AK from authentication
    String SK="xxx"; //SK from authentication
    String regionName="cn-north-4"; //region name of the service
    String httpUri = "/v1.0/ocr/id-card";
    String imgPath = "./data/id-card-demo.jpg"; //File path or URL of the image to be recognized.

    // Set params except image
    String sideKey = "side";
    String sideValue = "front";
    JSONObject params = new JSONObject();
    //params.put(sideKey, sideValue);

    try {
        HWOcrClientAKSK ocrClient=new HWOcrClientAKSK(regionName, AK, SK);
        HttpResponse response=ocrClient.RequestOcrServiceBase64(httpUri, imgPath, params);
        System.out.println(response);
        String content = IOUtils.toString(response.getEntity().getContent(), "utf-8");
        System.out.println(content);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

Table 3-1 Parameters in the Main function

Parameter	Description	Value
AK	Access Key ID (AK)	Log in to the **My Credential** page and choose **Access Keys** > **Add Access Key** to obtain it.
SK	Secret access key (SK)	Log in to the **My Credential** page and choose **Access Keys** > **Add Access Key** to obtain it.
regionName	Region where the service resides	cn-north-4
httpUri	Uniform resource identifier	/v1.0/ocr/id-card
imgPath	Image file path	./data/id-card-demo.jpg
sideKey	Optional parameter	side API request parameter of ID Card OCR. The options are as follows: • **front**: front side of the ID card with the profile • **back**: back side of the ID card with the national emblem

Parameter	Description	Value
sideValue	Value of the optional parameter	The value can be **front** or **back**.

Step 4: Calling the Service

Execute the **OCRDemo.java** file. If status code **200** is displayed, the program is executed. The recognition result of ID Card OCR is returned in JSON format, as shown below.

```
{
   "result": {
      "name": "xx",
      "sex": "Gender recognized from the image",
      "ethnicity": "Ethnicity recognized from the image",
      "birth": "1990-xx-xx",
      "address": "Address recognized from the image",
      "number": "3892011990012xxxxx"
   }
}
```

