Optical Character Recognition

API Reference

Issue	01
Date	2024-03-29

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

1.1 Overview

OCR provides services through open Application Programming Interfaces (APIs). You can obtain the inference result by accessing and calling APIs in real time. It helps you collect key data automatically and build an intelligent business system, thereby improving service efficiency. Before calling an API, ensure that the user network can access the Internet.

You can perform related operations based on the API description, syntax, parameter description, and examples provided in this document. For example, you can call the API for recognizing characters in general text, cards, or receipts. APIs vary depending on regions. For details, see **Endpoints**. For details about all operations supported by APIs, see **API Overview**.

Before calling an OCR API, ensure that you are familiar with OCR concepts.

OCR also provides software development kits (SDKs) for multiple programming languages. For details about how to use SDKs, see **SDK Reference**.

1.2 API Calling

OCR provides a broad range of Representational State Transfer (REST) APIs that you can call through HTTPS. For details about API calling, see **Calling APIs**.

Before calling an API, ensure that the user network can access the Internet.

OCR also provides software development kits (SDKs) for multiple programming languages. For details about how to use SDKs, see **SDK Reference**.

1.3 Endpoints

An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.

Table 1-1 lists the endpoints of OCR. Select an endpoint based on your service requirements.

Even if the service server is not in the region where the OCR service is deployed, the OCR service can still be used as long as the server can access the Internet. It is recommended that you select the closest region for low network latency and quick access.

Table 1-1 OCR endpoints

Region	Endpoint Region	Endpoint	Service
CN-Hong Kong	ap-southeast-1	ocr.ap-southeast-1.myhuaweicloud.com ocr.ap-southeast-1.myhuaweicloud.cn	Passport OCR, Myanmar ID Card OCR, Myanmar Driving License OCR, Hong Kong ID Card OCR, Macao Resident ID Card OCR, Exit-Entry Permit for Traveling to and from Hong Kong, Macao, and Taiwan OCR, Mainland Travel Permit for Hong Kong, Macao, and Taiwan Residents OCR, ID Document OCR, General Text OCR, and Vietnam ID Card OCR
AP-Bangkok	ap-southeast-2	ocr.ap-southeast-2.myhuaweicloud.com ocr.ap-southeast-2.myhuaweicloud.cn	Thailand ID Card OCR, Passport OCR, Web Image OCR, and Thailand Plate Number OCR
LA-Santiago	la-south-2	ocr.la-south-2.myhuaweicloud.com ocr.la-south-2.myhuaweicloud.cn	Chile ID Card OCR and Peru ID Card OCR
AP-Singapore	ap-southeast-3	ocr.ap-southeast-3.myhuaweicloud.com ocr.ap-southeast-3.myhuaweicloud.cn	Cambodian ID Card OCR and General Table OCR
AF-Johannesburg	af-south-1	ocr.af-south-1.myhuaweicloud.com ocr.af-south-1.myhuaweicloud.cn	Web Image OCR

1.4 Constraints and Limitations

For details, see the API description and **Constraints and Limitations**.

1.5 Concepts

- Account

  An account is created upon successful registration with Huawei Cloud and has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity. Do not directly use an account for routine management, but create users and assign them permissions for that.

- User

  A user is created on in IAM using an account to use cloud services. Each user has its own identity credentials (password and access keys).

  An IAM user can view the account ID and user ID on the **My Credentials** page of the console. The account name, username, and password will be required for API authentication.

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are accessible within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

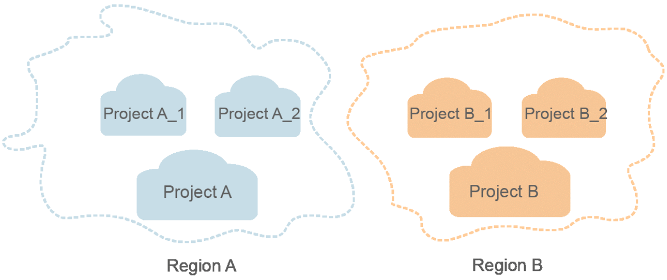
- AZ

  An AZ contains one or more physical data centers. It has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

  Projects isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each Huawei Cloud region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and purchase resources in the subprojects. Users can then be assigned permissions to access specific resources in the subprojects.

Figure 1-1 Project isolation model



2 API Overview

You can perform the operations described in **Table 2-1** with OCR APIs.

Table 2-1 API description

API	Description	Deployment Region
General Table	This API detects and extracts text from images of general tables and converts the text into JSON format. The returned results include two types of image area (words_region): text area (text) and table area (table). They also include table structures (rows and columns) and text information. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.	AP-Singapore (ap-southeast-3)
General Text	This API detects and extracts text from images and converts the text and coordinates into JSON format. It can be used in various scenarios, such as scanned files, electronic documents, books, receipts, and forms. English and Chinese are supported but support for traditional Chinese characters is limited. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.	CN-Hong Kong (ap-southeast-1)
Passport	This API detects and extracts text from the first page of images of passports and converts the text into a structured JSON format.	CN-Hong Kong (ap-southeast-1) AP-Bangkok (ap-southeast-2)

API	Description	Deployment Region
Web Image	This API detects and extracts text from web images and converts the text into a structured JSON format.	AP-Bangkok (ap-southeast-2)
Thailand ID Card	This API detects and extracts text from images of Thailand-issued national registration cards and converts the text into a structured format.	AP-Bangkok (ap-southeast-2)
Myanmar ID Card	This API detects and extracts text from images of Myanmar-issued national registration cards and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)
Cambodian ID Card	This API detects and extracts text from images of Cambodia-issued ID cards and converts the text into a structured format.	AP-Singapore (ap-southeast-3)
Myanmar Driving License	This API detects and extracts text from images of Myanmar-issued driver's licenses and converts the text into a structured JSON format.	CN-Hong Kong (ap-southeast-1)
Chile ID Card	This API detects and extracts text from images of Chile-issued ID cards and converts the text into JSON format.	LA-Santiago-(la-south-2)
Vietnam ID Card	This API detects and extracts text from images of Vietnam-issued ID cards and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)
Peru ID Card	This API detects and extracts text from images of Peru-issued identity cards and converts the text into a structured format.	LA-Santiago-(la-south-2)
Thailand Plate Number	This API detects and extracts license plate information from images of Thailand license plates and returns the license plate number and location.	AP-Bangkok (ap-southeast-2)
Hong Kong ID Card	This API detects and extracts text from images of Hong Kong (China)-issued ID cards and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)
Macao Resident ID Card	This API detects and extracts text from images of Macao (China)-issued resident ID cards and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)
Exit-Entry Travel Permit	This API detects and extracts text from images of exit-entry permits and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)

API	Description	Deployment Region
Chinese Mainland Travel Permit	This API detects and extracts text from images of mainland travel permits and converts the text into a structured format. This OCR service supports the following two types of mainland travel permits: mainland travel permit for Hong Kong (China) and Macao (China) residents and mainland travel permit for Taiwan (China) residents.	CN-Hong Kong (ap-southeast-1)
ID Document	This API detects and extracts text from images of identity documents issued by multiple countries and regions, such as ID cards, driving licenses, and passports, and converts the text into a structured format.	CN-Hong Kong (ap-southeast-1)

3 Calling APIs

3.1 Overview

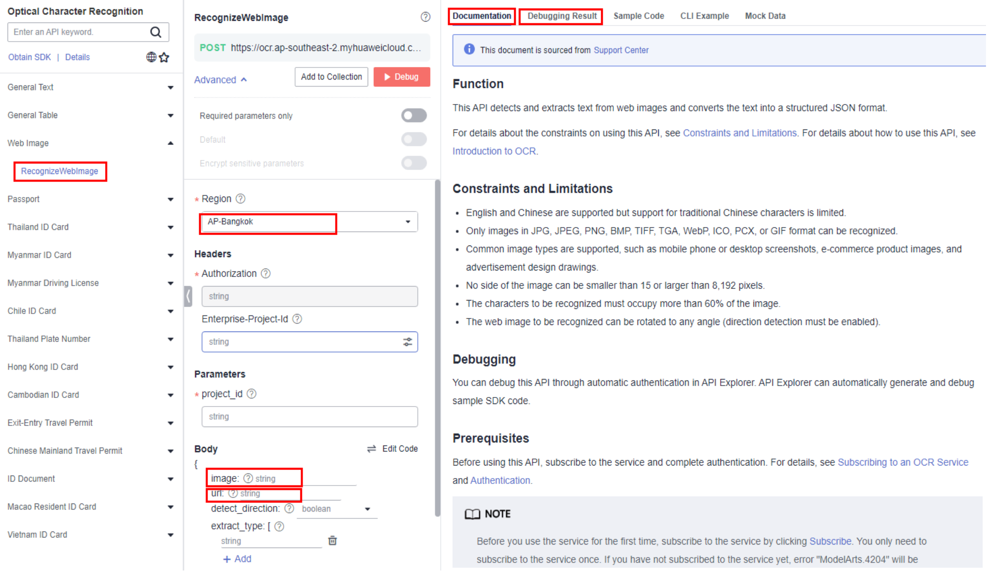
You can debug API calls using Postman or SDK. This involves subscribing to a service, configuring the environment, authenticating, and constructing a request to call the service.

1. Subscribing to a service: includes registration and login, IAM account creation (optional), and service subscription. For details, see **Subscribing to an OCR Service**.
2. Configuring the environment: You can choose to **download Postman** or **download an SDK**.
3. Authenticating: You can choose **Token-based Authentication** (Postman environment) or **AK/SK-based Authentication** (SDK environment). Select an authentication mode based on the environment configured in step 2.
4. Constructing a request. For details about how to construct a request in the Postman environment, see **Making an API Request**. For details about how to call an API in an SDK environment, see **Optical Character Recognition SDK Reference**.

Apart from the preceding two methods, you can also call APIs through **API Explorer online debugging**. The procedure is as follows:

1. The API Explorer page is displayed after a service is subscribed to by referring to **Subscribing to an OCR Service**.
2. Choose the subscribed service and the corresponding region for the service.
3. In the **Body** area, set **image** or **url**. Set other parameters based on site requirements. For details about the parameters, refer to the content displayed on the **Documentation** tab on the right of the page.
4. Click **Debug** and check the calling result on **Debugging Result** on the right.

Figure 3-1 API Explorer online debugging



This document is sourced from Support Center

Function

This API detects and extracts text from web images and converts the text into a structured JSON format.

For details about the constraints on using this API, see Constraints and Limitations. For details about how to use this API, see Introduction to OCR.

Constraints and Limitations

- English and Chinese are supported but support for traditional Chinese characters is limited.
- Only images in JPG, JPEG, PNG, BMP, TIFF, TGA, WebP, ICO, PCX, or GIF format can be recognized.
- Common image types are supported, such as mobile phone or desktop screenshots, e-commerce product images, and advertisement design drawings.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- The characters to be recognized must occupy more than 60% of the image.
- The web image to be recognized can be rotated to any angle (direction detection must be enabled).

Debugging

You can debug this API through automatic authentication in API Explorer. API Explorer can automatically generate and debug sample SDK code.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see Subscribing to an OCR Service and Authentication.

NOTE

Before you use the service for the first time, subscribe to the service by clicking Subscribe. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be

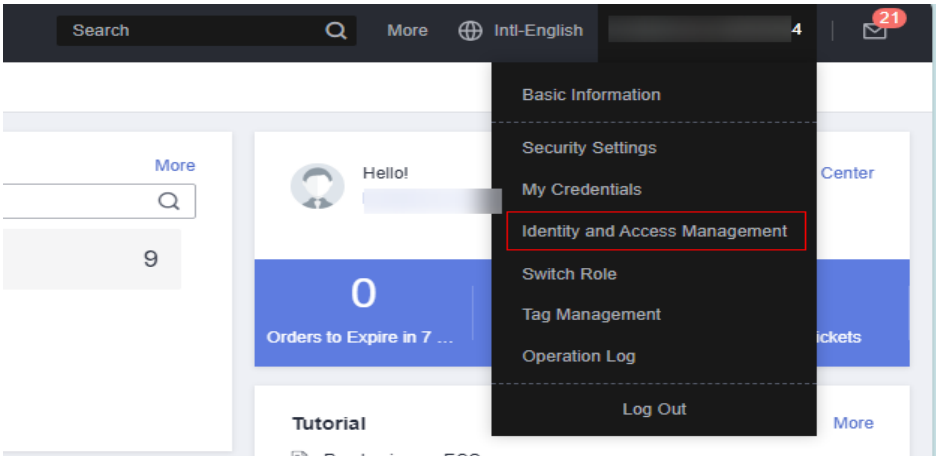
3.2 Subscribing to an OCR Service

Step 1 Register a Huawei Cloud account on the **Huawei Cloud official website** and use it to log in. The account cannot be in arrears or frozen.

Step 2 (Optional) Create an IAM user.

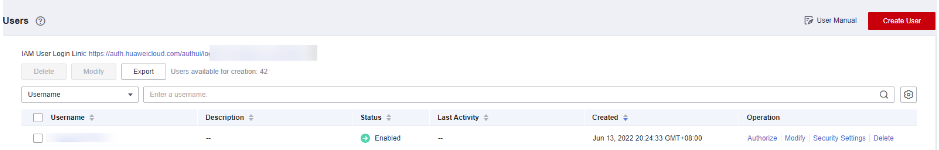
1. Log in to the Huawei Cloud **management console**, hover your cursor over the username in the upper right corner of the page, and choose **Identity and Access Management** from the drop-down list.

Figure 3-2 Identity and Access Management



2. On the **Users** page, click **Create User**. On the page displayed, set the username and password to create an IAM user.

Figure 3-3 Create User



Users

User Manual | Create User

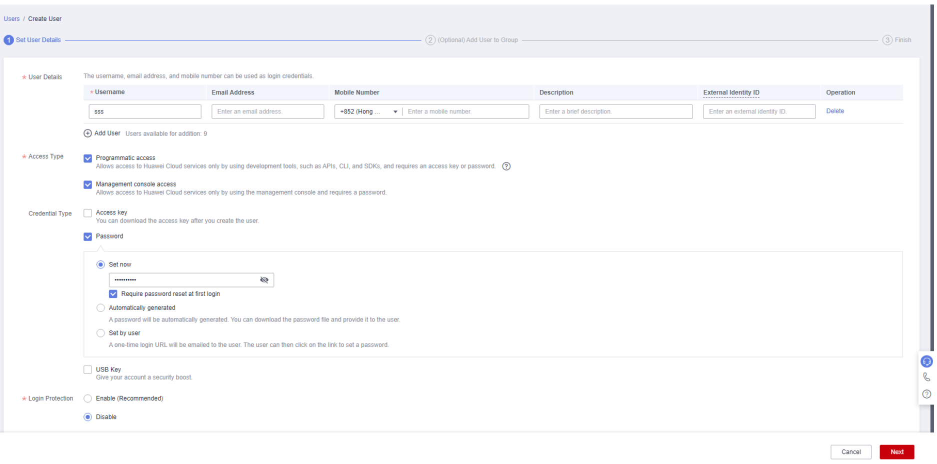
IAM User Login Link: https://auth.huaweicloud.com/authui/lo[redacted]

Delete | Modify | Export | Users available for creation: 42

Username ▾ | Enter a username.

□ Username	Description	Status	Last Activity	Created	Operation
□ [redacted]	--	Enabled	--	Jun 13, 2022 20:24:33 GMT+08:00	Authorize | Modify | Security Settings | Delete

Figure 3-4 Setting the username and password



Users / Create User

1 Set User Details — 2 (Optional) Add User to Group — 3 Finish

* User Details: The username, email address, and mobile number can be used as login credentials.

* Username	Email Address	Mobile Number	Description	External Identity ID	Operation
sss	Enter an email address.	+852 (Hong ... ▾ Enter a mobile number.	Enter a brief description.	Enter an external identity ID.	Delete

⊕ Add User Users available for addition: 9

* Access Type

☑ Programmatic access
Allows access to Huawei Cloud services only by using development tools, such as APIs, CLI, and SDKs, and requires an access key or password.

☑ Management console access
Allows access to Huawei Cloud services only by using the management console and requires a password.

Credential Type

□ Access key
You can download the access key after you create the user.

☑ Password

● Set now
••••••••••
☑ Require password reset at first login

○ Automatically generated
A password will be automatically generated. You can download the password file and provide it to the user.

○ Set by user
A one-time login URL will be emailed to the user. The user can then click on the link to set a password.

□ USB Key
Give your account a security boost.

* Login Protection

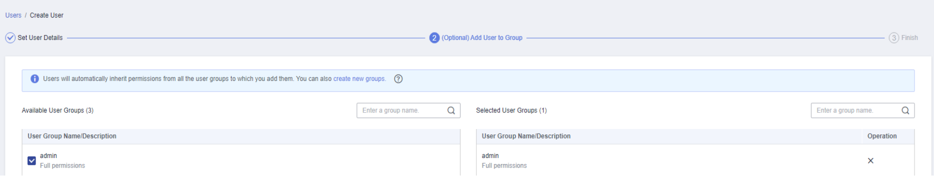
○ Enable (Recommended)

● Disable

Cancel | Next

3. Grant permissions to the IAM user. For details, see **Permissions Management**.

Figure 3-5 Adding the IAM user to a user group



Users / Create User

✓ Set User Details — 2 (Optional) Add User to Group — 3 Finish

Users will automatically inherit permissions from all the user groups to which you add them. You can also create new groups.

Available User Groups (3) | Enter a group name.

User Group Name/Description
☑ admin Full permissions

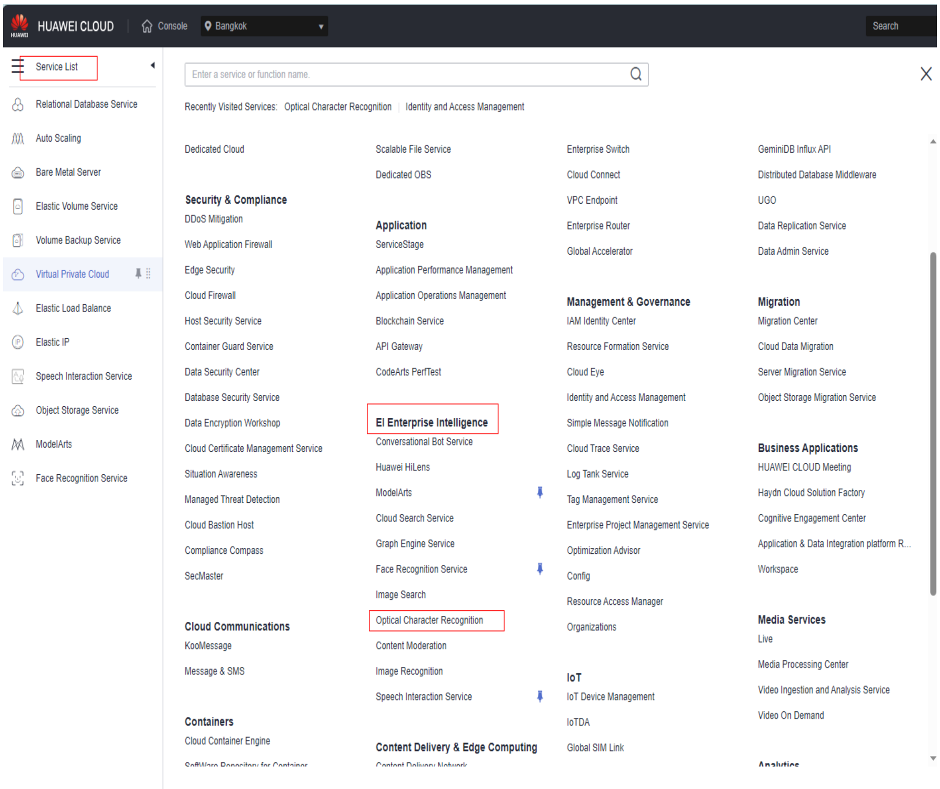
Selected User Groups (1) | Enter a group name.

User Group Name/Description	Operation
admin Full permissions	×

Step 3 Subscribe to a service.

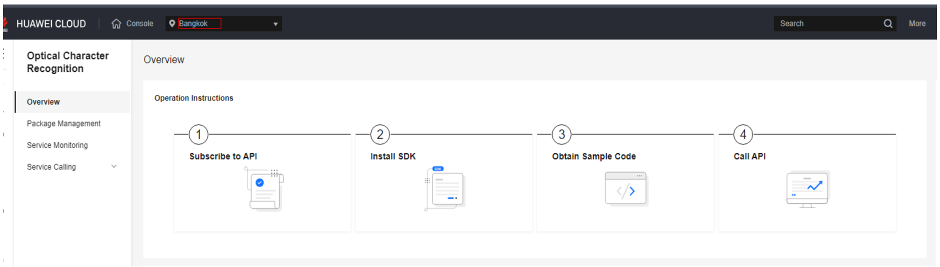
1. On the management console, click **Service List** and choose **EI Enterprise Intelligence** > **Optical Character Recognition** to access the **OCR console**.

Figure 3-6 Accessing the OCR console



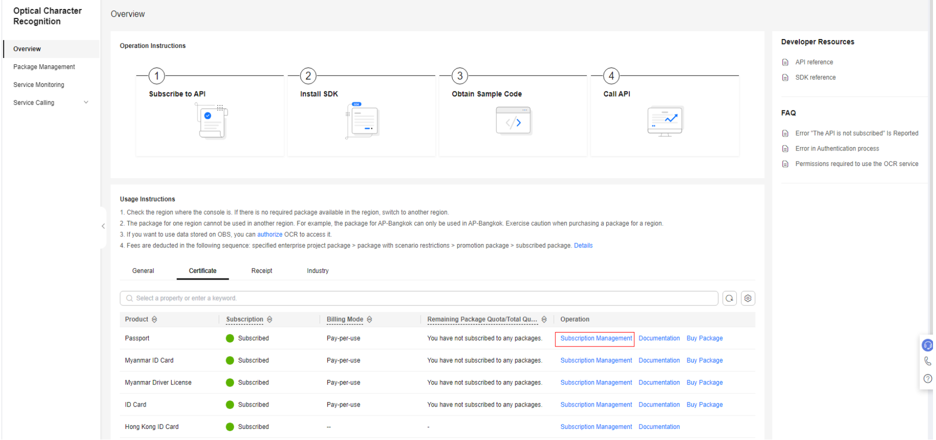
2. On the **Overview** page of the OCR console, select the endpoint where you want to subscribe to a service. For details about OCR service availability in different regions, see **Endpoints**.

Figure 3-7 Selecting an endpoint



3. Select a product type, for example, **General** and **Certificate**, and select your desired service to subscribe to it. Set **Billing Mode** to **Pay-per-use**.

Figure 3-8 Subscribing to a service

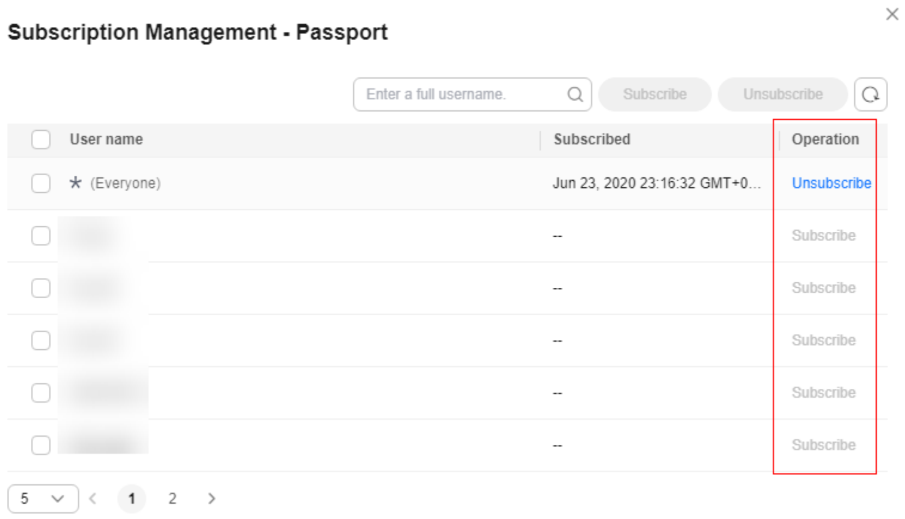


Usage Instructions

1. Check the region where the console is. If there is no required package available in the region, switch to another region.
2. The package for one region cannot be used in another region. For example, the package for AP-Bangkok can only be used in AP-Bangkok. Exercise caution when purchasing a package for a region.
3. If you want to use data stored on OBS, you can authorize OCR to access it.
4. Fees are deducted in the following sequence: specified enterprise project package > package with scenario restrictions > promotion package > subscribed package. Details

Product	Subscription	Billing Mode	Remaining Package Quota/Total Qu...	Operation
Passport	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
Myanmar ID Card	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
Myanmar Driver License	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
ID Card	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
Hong Kong ID Card	● Subscribed	--	-	Subscription Management Documentation

Figure 3-9 Subscription Management



Subscription Management - Passport

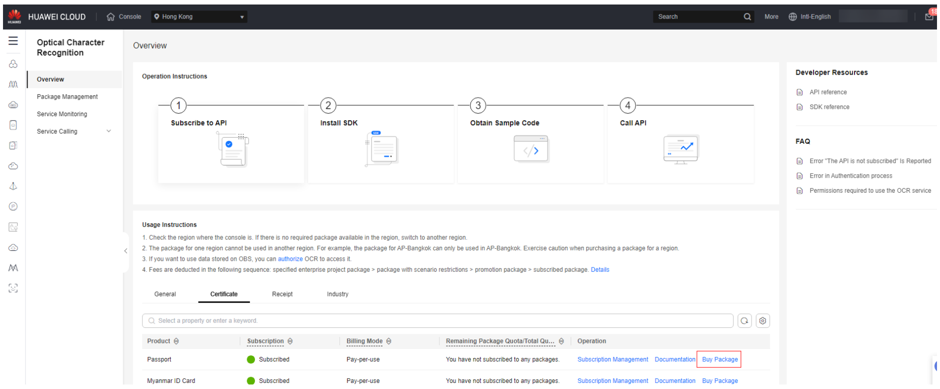
□	User name	Subscribed	Operation
□	* (Everyone)	Jun 23, 2020 23:16:32 GMT+0...	Unsubscribe
□		--	Subscribe
□		--	Subscribe
□		--	Subscribe
□		--	Subscribe
□		--	Subscribe

NOTE

- If a master account grants the **OCR FullAccess** permission to an IAM user, both the master account and the IAM user can subscribe to OCR services. IAM users without this permission cannot subscribe to OCR services on their own.
- If a master account only grants the **OCR ReadOnlyAccess** permission or no permission at all to an IAM user, only the master account (or IAM users with the **OCR FullAccess** permission) can subscribe to OCR services for that IAM user.

Alternatively, you can click **Buy Package** to subscribe to a service. The billing is on a package basis.

Figure 3-10 Buy Package



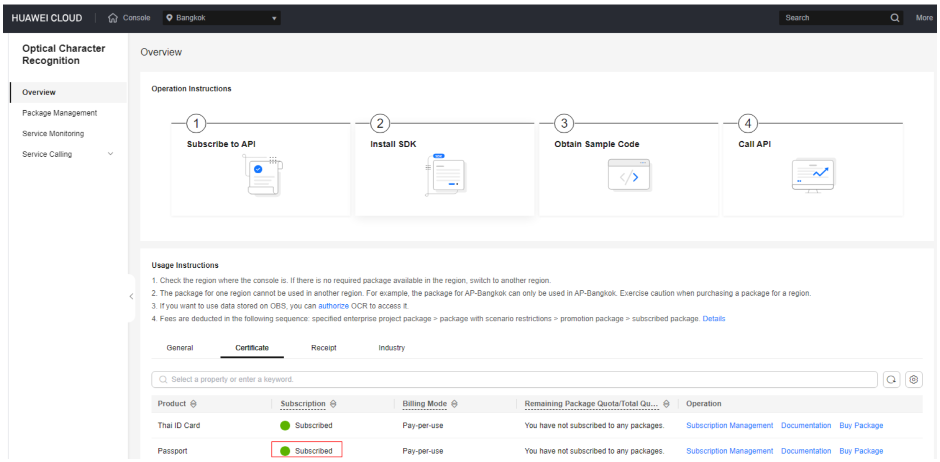
Usage Instructions

1. Check the region where the console is. If there is no required package available in the region, switch to another region.
2. The package for one region cannot be used in another region. For example, the package for AP-Bangkok can only be used in AP-Bangkok. Exercise caution when purchasing a package for a region.
3. If you want to use data stored on OBS, you can authorize OCR to access it.
4. Fees are deducted in the following sequence: specified enterprise project package > package with scenario restrictions > promotion package > subscribed package. Details

Product	Subscription	Billing Mode	Remaining Package Quota/Total Qu...	Operation
Passport	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
Myanmar ID Card	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package

4. Once the service is successfully subscribed to, check that the status is **Subscribed**.

Figure 3-11 Subscribed



Usage Instructions

1. Check the region where the console is. If there is no required package available in the region, switch to another region.
2. The package for one region cannot be used in another region. For example, the package for AP-Bangkok can only be used in AP-Bangkok. Exercise caution when purchasing a package for a region.
3. If you want to use data stored on OBS, you can authorize OCR to access it.
4. Fees are deducted in the following sequence: specified enterprise project package > package with scenario restrictions > promotion package > subscribed package. Details

Product	Subscription	Billing Mode	Remaining Package Quota/Total Qu...	Operation
Thai ID Card	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package
Passport	● Subscribed	Pay-per-use	You have not subscribed to any packages.	Subscription Management Documentation Buy Package

NOTE

- OCR services are billed in pay-per-use mode by default. In this billing mode, you only need to pay for what you use. If you have purchased a package, see **Billing** for package deduction rules. Packages cannot be refunded once being purchased.
- If you have not subscribed to an OCR service yet, an error message with error code "ModelArts.4204" will be displayed when you call the OCR API.
- If you want to use data stored on OBS, you can authorize OCR to access OBS. You only need to configure the authorization once. For details, see **Configuring Access Permissions of OBS**.

----End

3.3 Making an API Request

This section describes the structure of a REST API request and demonstrates how to call an API.

Request URI

A request URI is in the following format:

{URI-scheme} :// {endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages and frameworks require the request URI to be transmitted separately.

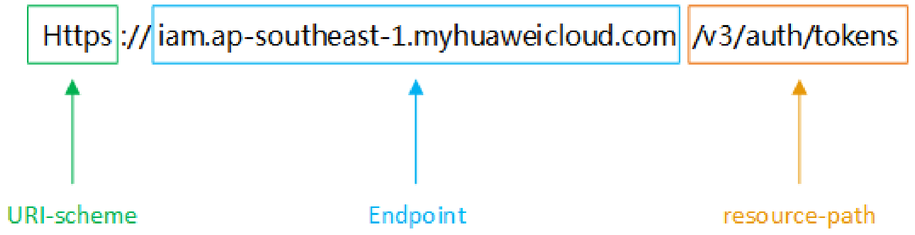
Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
endpoint	Domain name or IP address of the server bearing the REST service endpoint. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
resource-path	Path in which the requested resource is located, that is, the API access path.
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of *Parameter name=Parameter value*. For example, **limit=10** indicates that up to 10 data records will be displayed.

For example, to call the API in the **AP-Bangkok** region, use the endpoint **ocr.ap-southeast-2.myhuaweicloud.com** of the **AP-Bangkok** region and find the value **/v3/auth/tokens** of **resource-path** in **Obtaining a User Token**. The URI is as follows:

```
https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image
```

Figure 3-12 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

Table 3-2 HTTP-defined request methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

```
POST https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

- **Content-Type**: request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: user token. This field is required only for token-based API authentication. For details about the user token, see "Authentication".

The following provides an example request with a request header included.

```
POST https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image
Content-Type: application/json
x-auth-token: MIIaBgYJKoZIhvcNAQcC...
```

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers all content except the request header.

The request body varies depending on APIs. Some APIs do not require the request body, for example, the APIs requested using the GET and DELETE methods.

The request parameters and parameter description are available in the request. The following provides an example request with a body included.

```
POST https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image
Request Header:
```

```
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAgABwESAAMAA..."
}
```

If all data required for the API request is available, you can send the request to call the API through curl, Postman, or coding.

3.4 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair.

Token-based Authentication

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to a request to get permissions for calling the API.

NOTE

- A token is valid for 24 hours. When using a token for authentication, cache it to avoid frequent calling.
- If your Huawei Cloud account has been upgraded to a Huawei ID, you cannot obtain a token. You are advised to create an IAM user and obtain the user token.

When calling the API to obtain a user token, you must set **auth.scope** in the request body to **project**.

You can log in to the console and choose **My Credentials > API Credentials** to obtain the values of **username**, **domainname**, and **project name**. **password** indicates the user password.

- Pseudocode

```
POST https://iam.ap-southeast-2.myhuaweicloud.com/v3/auth/tokens //Uses obtaining the token in the AP-Bangkok region as an example.
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username", // IAM username
                    "password": "********", // Password
                    "domain": {
                        "name": "domainname" // Account name
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "project name" //Replace project name with the actual project name, for example, ap-southeast-1.
            }
        }
    }
}
```

- Java

```
package xxx; // Project path where the GetToken class is located

import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class GetToken {
    public static void main(String[] args) throws Exception {
        OkHttpClient client = new OkHttpClient().newBuilder().build();
        MediaType mediaType = MediaType.parse("application/json");
        String requestBody // Enter the correct account name, IAM username, and password.
            = "{\"auth\": {\"identity\": {\"methods\": [\"password\"],\"password\": {\"user\": {\"name\": \"********\",\"password\": \"********\",\"domain\": {\"name\": \"********\"}}}},\"scope\": {\"project\": {\"name\": \"ap-southeast-3\"}}}}";
        RequestBody body = RequestBody.create(requestBody, mediaType);
        Request request = new Request.Builder().url("https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens")
            .method("POST", body)
            .addHeader("Content-Type", "application/json")
            .build();
        Response response = client.newCall(request).execute();
        System.out.println(response.header("X-Subject-Token"));
    }
}
```

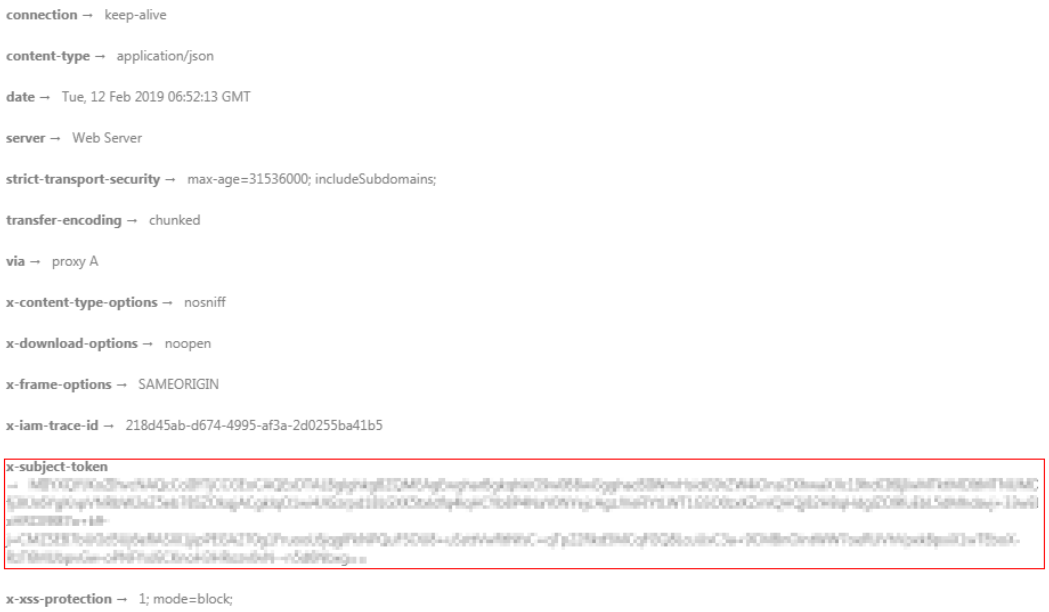
- Python

```
import requests
import json

url = "https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens"
payload = json.dumps({
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "********",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "projectname"
      }
    }
  }
})
headers = {
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)
```

```
print(response.headers["X-Subject-Token"])
```

As shown in the following figure, **x-subject-token** in the response header is the desired user token. This token can then be used to authenticate the calling of OCR APIs.

Figure 3-13 Response header for obtaining a user token



```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token
→ [redacted]
x-xss-protection → 1; mode=block;
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body less than or equal to 12 MB. For API requests with a larger body, perform token-based authentication.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign a request based on the signature algorithm or use a dedicated signature SDK to sign a request. For details about how to sign requests and use the signing SDK, see ***API Request Signing Guide***.

If no AKs/SKs have been generated, log in to the console and choose **My Credentials** page in the upper right corner. On the page that is displayed, choose **Access Keys** in the navigation pane on the left, and click **Create Access Key** to create an AK/SK.

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

For details about how to obtain the AK/SK, see **Obtaining the AK/SK**.

3.5 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code consists of multiple digits, which indicates the response status of a request. If the 2*xx* status code is returned, the API is called successfully. For details about the status code list, see **Status Codes**.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**. **Table 3-3** lists the fields in the response header.

Table 3-3 Response Header

Field	Description
Content-Length	Length of the response body. The unit is byte.
Date	Time when a request response is returned
Content-Type	MIME type of the response body

Response Body

The body of a response is returned in structured format as specified in the **Content-Type** header field. The response body transfers all content except the response header.

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

In the response body, **error_code** indicates the error code, and **error_msg** provides information about the error.

4 API

4.1 ID Document

Function

This API detects and extracts text from images of identity documents and converts the text into a structured format. These documents include ID cards, driving licenses, and passports from multiple countries and regions. **Table 4-1** lists the mapping between supported countries/regions and document types. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Table 4-1 Mapping between countries/regions and document types

Country/Region	Code	Document Type
Vietnam	VNM	PP, DL, and ID
India	IND	PP
Philippines	PHL	PP, DL, ID (UMID only)
Albania	ALB	PP, DL, and ID
Brazil	BRA	PP
Indonesia	IDN	PP
Malaysia	MYS	PP
Nigeria	NGA	PP
Pakistan	PAK	PP
Russia	RUS	PP (Only the international standard version is supported.)
Taiwan (China)	TWN	PP

Country/Region	Code	Document Type
Ukraine	UKR	PP
Thailand	THA	ID and PP
Chile	CHL	ID and PP
Hong Kong (China)	HKG	ID
• PP: passport • DL: driving license • ID: identification card, which is an identity card issued by a country or region, such as an ID card, voter registration card, and social security card.		

Constraints and Limitations

- Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 100 or larger than 8,192 pixels.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/id-document

Table 4-2 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.

Parameter	Mandatory	Description
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

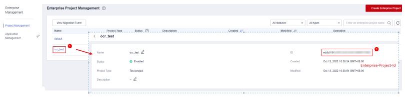
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-4 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 100 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
country_region	No	Array of strings	Code of the country or region where a certificate is issued. The code must be defined in ISO 3166-1 alpha-2 codes. This parameter is optional. You can enter one or more country/region codes. After this parameter is specified, the service identifies cards only in the specified country or region. If this parameter is left blank, all supported cards are identified. It is recommended that this field be filled in when the country/region is fixed or limited. For the list of supported countries and regions, see **Table 4-1**. **NOTE** This parameter is mandatory when Vietnamese is recognized.
id_type	No	Array of strings	Document type. This parameter is optional. One or more types of documents are supported. If this parameter is specified, the service identifies only the documents of the specified type. If this parameter is left blank, all types of documents are identified by default. You are advised to configure this parameter if the document type is known. The following document types are supported: ● **PP**: passport ● **DL**: driving license ● **ID**: identity card, which is an identity card issued by a country or region, such as an ID card, voter registration card, and social security card.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the portrait image (face image in the document). The value **true** indicates that the portrait image needs to be returned, and the value **false** indicates that the portrait image does not need to be returned.

Response Parameters

Status code: 200

Table 4-5 Response body parameter

Parameter	Type	Description
result	**IdDocumentItem** object	Recognition result This parameter is not included when the API fails to be called.

Table 4-6 IdDocumentItem

Parameter	Type	Description
country_region	String	Code of the country or region where a certificate is issued. The code must be defined in ISO 3166-1 alpha-2 codes. For the list of supported countries and regions, see **Table 4-1**.
id_type	String	Document type. The options are as follows: ● **PP**: passport ● **DL**: driving license ● **ID**: identity card, which is an identity card issued by a country or region, such as an ID card, voter registration card, and social security card.
side	String	Front or back of an identity document. Value options are as follows: ● **front**: front side of the document, which is typically the side that contains a portrait. ● **back**: **front** is returned if a document has only one side.

Parameter	Type	Description
first_name	String	First name
last_name	String	Last name
sex	String	Gender. The options are as follows: **M**: male; **F**: female; **X**: third gender.
nationality	String	Nationality of the document holder
birth_date	String	Date of birth, in YYYY-MM-DD format
issue_date	String	Date of issue, in YYYY-MM-DD format
expiry_date	String	Expiration date, in YYYY-MM-DD format
document_number	String	Document number
address	String	Contact address of the holder
issuing_authority	String	Issuing authority
portrait_image	String	Base64 code of the head portrait on the document, which is optional
confidence	Object	Field confidence. The value is a decimal ranging from 0 to 1. A larger value indicates more reliable recognition results.

Status code: 400

Table 4-7 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. This parameter is not included when the API is successfully called.
error_msg	String	Error message when the API call fails. This parameter is not included when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, ID Document OCR is deployed in the **CN-Hong Kong** region. The endpoint is **ocr.ap-southeast-1.myhuaweicloud.com** or **ocr.ap-southeast-1.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/id-document**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the image Base64 string.)
```
POST https://{endpoint}/v2/{project_id}/ocr/id-document

{
  "image" : "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "country_region" : "ALB",
  "id_type" : "PP"
}
```
- Request example (Method 2: Use the image URL.)
```
POST https://{endpoint}/v2/{project_id}/ocr/id-document

{
  "url" : "https://BucketName.obs.xxxx.com/ObjectName",
  "country_region" : "ALB",
  "id_type" : "PP"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "country_region" : "ALB",
    "id_type" : "PP",
    "side" : "front"
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "AIS.0103",
  "error_msg" : "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the certificate image for recognition.
```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class RecognizeIdDocumentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeIdDocumentRequest request = new RecognizeIdDocumentRequest();
        IdDocumentRequestBody body = new IdDocumentRequestBody();
        List<String> listbodyIdType = new ArrayList<>();
        listbodyIdType.add("PP");
        List<String> listbodyCountryRegion = new ArrayList<>();
        listbodyCountryRegion.add("ALB");
        body.withIdType(listbodyIdType);
        body.withCountryRegion(listbodyCountryRegion);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeIdDocumentResponse response = client.recognizeIdDocument(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the certificate image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;
import java.util.List;
import java.util.ArrayList;

public class RecognizeIdDocumentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeIdDocumentRequest request = new RecognizeIdDocumentRequest();
        IdDocumentRequestBody body = new IdDocumentRequestBody();
        List<String> listbodyIdType = new ArrayList<>();
        listbodyIdType.add("PP");
        List<String> listbodyCountryRegion = new ArrayList<>();
        listbodyCountryRegion.add("ALB");
        body.withIdType(listbodyIdType);
        body.withCountryRegion(listbodyCountryRegion);
        body.withUrl("https://BucketName.obs.xxxx.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeIdDocumentResponse response = client.recognizeIdDocument(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the certificate image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeIdDocumentRequest()
        listIdTypebody = [
            "PP"
        ]
        listCountryRegionbody = [
            "ALB"
        ]
        request.body = IdDocumentRequestBody(
            id_type=listIdTypebody,
            country_region=listCountryRegionbody,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_id_document(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the certificate image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeIdDocumentRequest()
        listIdTypebody = [
            "PP"
        ]
        listCountryRegionbody = [
            "ALB"
        ]
        request.body = IdDocumentRequestBody(
            id_type=listIdTypebody,
            country_region=listCountryRegionbody,
            url="https://BucketName.obs.xxxx.com/ObjectName"
        )
        response = client.recognize_id_document(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the certificate image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeIdDocumentRequest{}
    var listIdTypebody = []string{
        "PP",
    }
    var listCountryRegionbody = []string{
        "ALB",
    }
    imageIdDocumentRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.IdDocumentRequestBody{
        IdType: &listIdTypebody,
        CountryRegion: &listCountryRegionbody,
        Image: &imageIdDocumentRequestBody,
    }
    response, err := client.RecognizeIdDocument(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the certificate image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeIdDocumentRequest{}
    var listIdTypebody = []string{
        "PP",
    }
    var listCountryRegionbody = []string{
        "ALB",
    }
    urlIdDocumentRequestBody:= "https://BucketName.obs.xxxx.com/ObjectName"
    request.Body = &model.IdDocumentRequestBody{
        IdType: &listIdTypebody,
        CountryRegion: &listCountryRegionbody,
        Url: &urlIdDocumentRequestBody,
    }
    response, err := client.RecognizeIdDocument(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.2 General Table

Function

This API detects and extracts text from images of general tables and converts the text into JSON format. The returned results include two types of image area (words_region): text area (text) and table area (table). They also include table structures (rows and columns) and text information. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- The area to be recognized must occupy more than 80% of the image. When scanning a table, ensure that the entire table and its surrounding area are included in the image.
- An image can be rotated to any angle.
- Text in images with complex backgrounds (such as outdoor scenery or anti-counterfeit watermarks) or distorted table lines cannot be recognized.
- English and Chinese are supported but support for traditional Chinese characters is limited.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/general-table

Table 4-8 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained from **Obtaining a Project ID**.

Request Parameters

Table 4-9 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

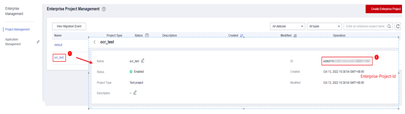
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description – ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-10 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data. ● The URL cannot contain Chinese characters. If Chinese characters exist, they must be encoded using UTF-8.
return_text_location	No	Boolean	Whether to return coordinates of text blocks and cells. Value options are as follows: ● **true**: Coordinates of text blocks and cells will be returned. ● **false**: Coordinates of text blocks and cells will not be returned. If this parameter is not specified, **false** is used by default.

Parameter	Mandatory	Type	Description
return_char_location	No	Boolean	Coordinate information of a single character. The options are as follows: ● **true**: The coordinates of a single character will be returned. ● **false**: The coordinates of a single character will not be returned. If this parameter is not specified, **false** is used by default. If this parameter is set to **true**, **return_text_location** must be **true**.
return_confidence	No	Boolean	Whether the confidence will be returned. The options are as follows: ● **true**: The confidence will be returned. ● **false**: The confidence will not be returned. If this parameter is not specified, **false** is used by default. In this case, the confidence will not be returned.
return_excel	No	Boolean	Whether to return the Base64-encoded field for converting a table into a Microsoft Excel file. The options are as follows: ● **true**: The Base64-encoded **excel** field will be returned. ● **false**: The Base64-encoded **excel** field will not be returned. The default value is **false**. You can use the Python function **base64.b64decode** to decode the returned Excel code and save it as an .xlsx file.
return_rectification_matrix	No	Boolean	The options are as follows: ● **true**: The perspective transformation matrix will be returned. ● **false**: The perspective transformation matrix will not be returned. If this parameter is not specified, **false** is used by default. In this case, the perspective transformation matrix will not be returned.

Parameter	Mandatory	Type	Description
with_borders	No	Boolean	The options are as follows: ● **true**: The input image contains only bordered tables, and only such tables are recognized. ● **false**: The input image may contain borderless tables, and both bordered and borderless tables are recognized. If this parameter is not specified, the default value **false** is used. If the input image contains only bordered tables, set this parameter to **true** to achieve more accurate recognition results.

Response Parameters

NOTE

The status code may vary depending on the recognition results. For example, **200** indicates that the API is successfully called, and **400** indicates that the API fails to be called. The following describes the status codes and corresponding response parameters.

Status code: 200

Table 4-11 Response body parameter

Parameter	Type	Description
result	**GeneralTableResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-12 GeneralTableResult

Parameter	Type	Description
words_region_count	Integer	Number of text areas
words_region_list	Array of **WordsRegionList** objects	List of recognition results in text areas. The output sequence is from left to right and from top to bottom.

Parameter	Type	Description
excel	String	The table image is converted into the Base64 code of the Excel file. The text and table in the image are written into the Excel file by position. Decode the returned code using base64.b64decode and save it as an .xlsx file.

Table 4-13 WordsRegionList

Parameter	Type	Description
type	String	Type of the text identification area. The options are as follows: ● **text**: text recognition area ● **table**: table recognition area
words_block_count	Integer	Number of text blocks recognized in a sub-area
words_block_list	Array of **GeneralTableWordsBlockList** objects	List of text blocks recognized in a sub-area. The output sequence is from left to right and from top to bottom.

Table 4-14 GeneralTableWordsBlockList

Parameter	Type	Description
words	String	Recognition result of a text block
confidence	Float	Average confidence of fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
location	Array<Array<Integer>>	Text block location information, in list format, indicating the X and Y coordinates of the four vertices in a text block. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.
words_list	Array of **WordsListItem** objects	List of the character blocks in a cell. The output sequence is from left to right and from top to bottom. This parameter is available only when the input parameter **return_text_location** is set to **true**.

Parameter	Type	Description
rows	Array of integers	Rows occupied by text. The values start from 0 and are displayed in a list. The data type is **Integer**. This parameter is valid only in table recognition areas, that is, this parameter is valid only when **type** is **table**.
columns	Array of integers	Columns occupied by text. The values start from 0 and are displayed in a list. The data type is **Integer**. This parameter is valid only in table recognition areas, that is, this parameter is valid only when **type** is **table**.
cell_location	Array<Array<Integer>>	Cell position information, in list format, indicating the X and Y coordinates of the four vertices in a cell. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.

Table 4-15 WordsListIem

Parameter	Type	Description
words	String	Recognition result of a text block
confidence	Float	Average confidence of fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
location	Array<Array<Integer>>	Text block location information, in list format, indicating the X and Y coordinates of the four vertices in a text block. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.
char_list	Array of **CharListIem** objects	List of the character blocks in a cell. The output sequence is from left to right and from top to bottom. This parameter is available only when the input parameters **return_text_location** and **return_char_location** are both set to **true**.

Table 4-16 CharListIem

Parameter	Type	Description
char	String	Recognition result of a single character

Parameter	Type	Description
char_confidence	Float	Confidence of a single character. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
char_location	Array<Array<Integer>>	Location information of a single character, in list format, indicating the X and Y coordinates of the four vertices in a text block. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.

Status code: 400

Table 4-17 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails. This parameter is not included when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, if General Table OCR is deployed in the **AP-Singapore** region, the endpoint is **ocr.ap-southeast-3.myhuaweicloud.com** or **ocr.ap-southeast-3.myhuaweicloud.cn**, and the request URL is **https://ocr.ap-southeast-3.myhuaweicloud.com/v2/{project_id}/ocr/general-table**. **project_id** indicates the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Authentication**.

- Transfer the Base64 code of a table image for recognition and does not return the confidence.

```
POST https://{endpoint}/v2/{project_id}/ocr/general-table
 Request Header:
 Content-Type: application/json
 X-Auth-Token: 
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
 Request Body:
 {
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAg...",
    "return_confidence":false
  }
```

- Transfer the URL of a table image for recognition and does not return the confidence.

```
POST https://{endpoint}/v2/{project_id}/ocr/general-table
 Request Header:
 Content-Type: application/json
 X-Auth-Token: 
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
 Request Body:
 {
     "url":"https://BucketName.obs.xxxx.com/ObjectName",
     "return_confidence":false
  }
```

Example Response

Status code: 200

Example of a successful response

```
{
  "result": {
    "words_region_count": 2,
    "words_region_list" : [ {
      "type" : "text",
      "words_block_count" : 1,
      "words_block_list" : [ {
        "words": "Text block 1 recognized in the text area",
        "confidence" : 0.9991
      } ]
    }, {
      "type" : "table",
      "words_block_count" : 2,
      "words_block_list" : [ {
        "words": "Text block 1 recognized in the table area",
        "confidence" : 0.9942,
        "rows" : [ 0 ],
        "columns" : [ 0 ]
      }, {
        "words": "Text block 2 recognized in the table area",
        "confidence" : 0.914,
        "rows" : [ 0 ],
        "columns" : [ 1, 2 ]
      } ]
    } ]
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "result" : {
    "error_code" : "AIS.0103",
    "error_msg" : "The image size does not meet the requirements."
  }
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of a table image for recognition and does not return the confidence.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeGeneralTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeGeneralTableRequest request = new RecognizeGeneralTableRequest();
        GeneralTableRequestBody body = new GeneralTableRequestBody();
        body.withReturnConfidence(false);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeGeneralTableResponse response = client.recognizeGeneralTable(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of a table image for recognition and does not return the confidence.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeGeneralTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeGeneralTableRequest request = new RecognizeGeneralTableRequest();
        GeneralTableRequestBody body = new GeneralTableRequestBody();
        body.withReturnConfidence(false);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeGeneralTableResponse response = client.recognizeGeneralTable(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of a table image for recognition and does not return the confidence.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeGeneralTableRequest()
        request.body = GeneralTableRequestBody(
            return_confidence=False,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_general_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of a table image for recognition and does not return the confidence.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeGeneralTableRequest()
        request.body = GeneralTableRequestBody(
            return_confidence=False,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_general_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of a table image for recognition and does not return the confidence.

```
package main

```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeGeneralTableRequest{}
    returnConfidenceGeneralTableRequestBody:= false
    imageGeneralTableRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.GeneralTableRequestBody{
        ReturnConfidence: &returnConfidenceGeneralTableRequestBody,
        Image: &imageGeneralTableRequestBody,
    }
    response, err := client.RecognizeGeneralTable(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of a table image for recognition and does not return the confidence.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeGeneralTableRequest{}
    returnConfidenceGeneralTableRequestBody:= false
    urlGeneralTableRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.GeneralTableRequestBody{
        ReturnConfidence: &returnConfidenceGeneralTableRequestBody,
        Url: &urlGeneralTableRequestBody,
    }
    response, err := client.RecognizeGeneralTable(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.3 General Text

Function

This API detects and extracts text from images and converts the text and coordinates into JSON format. It can be used in various scenarios, such as scanned files, electronic documents, books, receipts, and forms.

English and Chinese are supported but support for traditional Chinese characters is limited. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, GIF, TIFF, WebP, PCX, ICO, or PSD format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- The area to be recognized must occupy more than 80% of the image. When scanning a table, ensure that all text and its surrounding area are included in the image.
- An image can be rotated to any angle.
- Text in images with complex backgrounds (such as outdoor scenery or anti-counterfeit watermarks) or distorted text cannot be recognized.
- Supported languages: Chinese, English, some traditional Chinese, Malay, Ukrainian, Hindi, Russian, Vietnamese, Indonesian, Thai, Arabic, German, Latin, French, Italian, Spanish, Portuguese, Romanian, Polish Amharic, Japanese, Korean, Turkish, Norwegian, Danish, Swedish, and Khmer.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/general-text

Table 4-18 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained from **Obtaining a Project ID**.

Request Parameters

Table 4-19 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

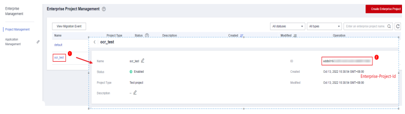
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: - If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. - If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. - If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-20 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, GIF, TIFF, WebP, PCX, ICO, or PSD format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data. ● The URL cannot contain Chinese characters. If Chinese characters exist, they must be encoded using UTF-8.

Parameter	Mandatory	Type	Description
detect_direction	No	Boolean	Whether to align the tilted image. The options are as follows: - **true**: The tilted image will be aligned. - **false**: The tilted image will not be aligned. An image tilted to any angle can be aligned. If this parameter is not specified, **false** is used by default. If the image to be recognized is tilted, you are advised to set this parameter to **true**.
quick_mode	No	Boolean	Whether to enable the quick mode. For a single-line text image (the image contains only one line of text and the text area occupies more than 50% of the image), the recognition results can be returned more quickly when this quick mode is enabled. The options are as follows: - **true**: The quick mode will be enabled. - **false**: The quick mode will be disabled. If this parameter is not specified, **false** is used by default. In this case, the quick mode will be disabled.
character_mode	No	Boolean	Whether to enable the single-character mode. The options are as follows: - **true**: The single-character mode is enabled. - **false**: The single-character mode is disabled. If this parameter is not transferred, the default value **false** is used, and information about a single character that occupies a text line is not returned.

Parameter	Mandatory	Type	Description
language	No	String	Language. If this parameter is not specified, Chinese and English will be used by default. The options are as follows: ● **auto**: automatic language classification ● **ms**: Malay ● **uk**: Ukrainian ● **hi**: Hindi ● **ru**: Russian ● **vi**: Vietnamese ● **id**: Indonesian ● **th**: Thai ● **zh**: Chinese and English ● **ar**: Arabic ● **de**: German ● **la**: Latin ● **fr**: French ● **it**: Italian ● **es**: Spanish ● **pt**: Portuguese ● **ro**: Romanian ● **pl**: Polish ● **am**: Amharic ● **ja**: Japanese ● **ko**: Korean ● **tr**: Turkish ● **no**: Norwegian ● **da**: Danish ● **sv**: Swedish ● **km**: Khmer
single_orientation_mode	No	Boolean	Whether to enable the single direction mode. The options are as follows: ● **true**: The single direction mode is enabled. ● **false**: The single direction mode is disabled. If this parameter is not specified, **false** is used by default. In this case, the fields in the image are recognized as in multiple directions by default.

Response Parameters

NOTE

The status code may vary depending on the recognition results. For example, **200** indicates that the API is successfully called, and **400** indicates that the API fails to be called. The following describes the status codes and corresponding response parameters.

Status code: 200

Table 4-21 Response body parameter

Parameter	Type	Description
result	**GeneralTextResult** object	Recognition result This parameter is not returned when the API fails to be called.

Table 4-22 GeneralTextResult

Parameter	Type	Description
direction	Float	Image direction ● This parameter is valid only when **detect_direction** is set to **true**. The anti-clockwise rotation angle of an image is returned. The value ranges from 0 to 359. ● When **detect_direction** is set to **false**, the value of this parameter is **-1**.
words_block_count	Integer	Number of detected text blocks
words_block_list	Array of **GeneralTextWordsBlockList** objects	List of recognized text blocks. The output sequence is from left to right and from top to bottom.

Table 4-23 GeneralTextWordsBlockList

Parameter	Type	Description
words	String	Recognition result of a text block

Parameter	Type	Description
location	Array<Array<Integer>>	List of location information about a text block, including the 2D coordinates (x, y) of four vertexes in the text area, where the coordinate origin is the upper-left corner of the image, the X axis is horizontal, and the Y axis is vertical.
confidence	Float	Confidence of a recognized text block
char_list	Array of **GeneralTextCharList** objects	Single-character recognition list corresponding to a text block. The output sequence is from left to right and from top to bottom.

Table 4-24 GeneralTextCharList

Parameter	Type	Description
char	String	Recognition result of a single character
char_location	Array<Array<Integer>>	List of location information about a single character, including the 2D coordinates (x, y) of four vertexes in the character area, where the coordinate origin is the upper-left corner of the image, the X axis is horizontal, and the Y axis is vertical.
char_confidence	Float	Confidence of a recognized character

Status code: 400

Table 4-25 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, if General Text OCR is deployed in the **CN-Hong Kong** region, the endpoint is **ocr.ap-southeast-1.myhuaweicloud.com** or **ocr.ap-southeast-1.myhuaweicloud.cn**, and the request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/general-text**. **project_id** indicates the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Authentication**.

- Transfer the Base64 code of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
POST https://{endpoint}/v2/{project_id}/ocr/general-text
 Request Header:
 Content-Type: application/json
 X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
 Request Body:
 {
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
    "detect_direction":false,
    "quick_mode":false
  }
```

- Transfer the URL of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
POST https://{endpoint}/v2/{project_id}/ocr/general-text
 Request Header:
 Content-Type: application/json
 X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
 Request Body:
 {
     "url":"https://BucketName.obs.xxxx.com/ObjectName",
     "detect_direction":false,
     "quick_mode":false
  }
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "direction" : 67.6506,
    "words_block_count" : 1,
    "words_block_list" : [ {
      "words" : "Word",
      "confidence" : 0.9999,
      "location" : [ [ 517, 447 ], [ 540, 504 ], [ 505, 518 ], [ 482, 461 ] ],
      "char_list" : [ {
        "char" : "Character",
        "char_location" : [ [ 517, 447 ], [ 530, 479 ], [ 495, 493 ], [ 482, 461 ] ],
        "char_confidence" : 0.9999
      }, {
        "char" : "Character",
        "char_location" : [ [ 530, 479 ], [ 540, 504 ], [ 505, 518 ], [ 495, 493 ] ],
        "char_confidence" : 0.9999
      } ]
    } ]
  }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeGeneralTextSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeGeneralTextRequest request = new RecognizeGeneralTextRequest();
        GeneralTextRequestBody body = new GeneralTextRequestBody();
        body.withQuickMode(false);
        body.withDetectDirection(false);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeGeneralTextResponse response = client.recognizeGeneralText(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeGeneralTextSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeGeneralTextRequest request = new RecognizeGeneralTextRequest();
        GeneralTextRequestBody body = new GeneralTextRequestBody();
        body.withQuickMode(false);
        body.withDetectDirection(false);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeGeneralTextResponse response = client.recognizeGeneralText(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeGeneralTextRequest()
        request.body = GeneralTextRequestBody(
            quick_mode=False,
            detect_direction=False,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_general_text(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeGeneralTextRequest()
        request.body = GeneralTextRequestBody(
            quick_mode=False,
            detect_direction=False,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_general_text(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeGeneralTextRequest{}
    quickModeGeneralTextRequestBody:= false
    detectDirectionGeneralTextRequestBody:= false
    imageGeneralTextRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.GeneralTextRequestBody{
        QuickMode: &quickModeGeneralTextRequestBody,
        DetectDirection: &detectDirectionGeneralTextRequestBody,
        Image: &imageGeneralTextRequestBody,
    }
    response, err := client.RecognizeGeneralText(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the image for recognition. During the recognition, the tilt angle of the image is not verified, and the quick mode is disabled.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeGeneralTextRequest{}
    quickModeGeneralTextRequestBody:= false
    detectDirectionGeneralTextRequestBody:= false
    urlGeneralTextRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.GeneralTextRequestBody{
        QuickMode: &quickModeGeneralTextRequestBody,
        DetectDirection: &detectDirectionGeneralTextRequestBody,
        Url: &urlGeneralTextRequestBody,
    }
    response, err := client.RecognizeGeneralText(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.4 Web Image

Function

This API detects and extracts text from web images and converts the text into a structured JSON format.

For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- English and Chinese are supported but support for traditional Chinese characters is limited.
- Only images in JPG, JPEG, PNG, BMP, TIFF, TGA, WebP, ICO, PCX, or GIF format can be recognized.
- Common image types are supported, such as mobile phone or desktop screenshots, e-commerce product images, and advertisement design drawings.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- The characters to be recognized must occupy more than 60% of the image.
- The web image to be recognized can be rotated to any angle (direction detection must be enabled).

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/web-image

Table 4-26 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained from **Obtaining a Project ID**.

Request Parameters

Table 4-27 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-28 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, JPEG, PNG, BMP, TIFF, TGA, WebP, ICO, PCX, or GIF format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.

Parameter	Mandatory	Type	Description
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data. ● The URL cannot contain Chinese characters. If Chinese characters exist, they must be encoded using UTF-8.
detect_direction	No	Boolean	Whether to align the tilted image. The options are as follows: ● **true**: The tilted image will be aligned. ● **false**: The tilted image will not be aligned. An image tilted to any angle can be aligned. If this parameter is not specified, **false** is used by default. If the image to be recognized is tilted, you are advised to set this parameter to **true**.
extract_type	No	Array of strings	Structured data extraction parameter list. Currently, only the image width and height are supported. The input parameter value of the image width and height is **image_size**. If this parameter is not set or is deleted, this parameter will not be used.
detect_font	No	Boolean	The value is of the Boolean type. If this parameter is not specified, slice fonts are not detected by default. If this parameter is set to **True**, the slice font type is detected and the five most similar font names are returned.

Parameter	Mandatory	Type	Description
detect_text_direction	No	Boolean	The value is of the Boolean type. If this parameter is not transferred, the default value **True** is used, indicating that the text direction of each field is detected. If this parameter is set to **False**, the text direction is not detected. If all text in the image faces up, you are advised to set this parameter to **False**.

Response Parameters

NOTE

The status code may vary depending on the recognition results. For example, **200** indicates that the API is successfully called, and **400** indicates that the API fails to be called. The following describes the status codes and corresponding response parameters.

Status code: 200

Table 4-29 Response body parameter

Parameter	Type	Description
result	**WebImageResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-30 WebImageResult

Parameter	Type	Description
words_block_count	Integer	This parameter is not included when the API fails to be called.
words_block_list	Array of **WebImageWordsBlockList** objects	List of text blocks to be recognized. The output sequence is from left to right and from top to bottom.
extracted_data	**WebImageExtractedData** object	Structured JSON results extracted. The key value in the dictionary is the same as the value of **extract_type** in the input parameter list. Currently, only the contact (**contact_info**) and image size (**image_size**) can be extracted. If **extract_type** is left blank or missing, no information is extracted.

Table 4-31 WebImageWordsBlockList

Parameter	Type	Description
words	String	Recognition result of a text block
confidence	Float	Confidence of related fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
location	Array<Array<Integer>>	List of location information about a text block, including the 2D coordinates (x, y) of four vertexes in the text area, where the coordinate origin is the upper-left corner of the image, the X axis is horizontal, and the Y axis is vertical.
font_list	Array of strings	Font type of a text block, in list format, indicating the font type closest to the font of the text in a text block.
font_scores	Array of numbers	Probability of the font type to which a text block belongs, in list format, corresponding to font_list, indicating the probability that the text in a text block belongs to a font type.

Table 4-32 WebImageExtractedData

Parameter	Type	Description
contact_info	**WebImageContactInfo** object	Extracted contact information, including the name, phone number, province, city, and detailed address. If **extract_type** does not contain this parameter, this parameter is not included in the response.
image_size	**WebImageImageSize** object	Width and height of an image. If **extract_type** does not contain this parameter, this parameter is not included in the response.

Table 4-33 WebImageContactInfo

Parameter	Type	Description
name	String	Name, which is returned when **contact_info** is specified
phone	String	Contact phone number, which is returned when **contact_info** is specified

Parameter	Type	Description
province	String	Province, which is returned when **contact_info** is specified
city	String	City, which is returned when **contact_info** is specified
district	String	County or district, which is returned when **contact_info** is specified
detail_address	String	Detailed address (excluding the province, city, and county or district), which is returned when **contact_info** is specified

Table 4-34 WebImageImageSize

Parameter	Type	Description
height	Integer	Image height, which is returned when **image_size** is specified
width	Integer	Image width, which is returned when **image_size** is specified

Status code: 400

Table 4-35 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails. This parameter is not included when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Web Image OCR is deployed in the **AP-Bangkok** region. The endpoint is **ocr.ap-southeast-2.myhuaweicloud.com** or **ocr.ap-southeast-2.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Authentication**.

- Transfer the Base64 code of a web image for recognition.

```
POST https://{endpoint}/v2/{project_id}/ocr/web-image
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
   "image":"/9j/4AAQSkZJRgABAgEASABIAAD/..."
}
```

- Transfer the URL of a web image for recognition.

```
POST https://{endpoint}/v2/{project_id}/ocr/web-image
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.xxxx.com/ObjectName"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result": {
      "words_block_count": 2,
      "words_block_list": [
          {
              "words": "Text block 1",
              "confidence": 0.9950,
              "location": [
                  [13, 476],
                  [91, 332],
                  [125, 351],
                  [48, 494]
              ]
          },
          {
              "words": "Text block 2",
              "confidence": 0.9910,
              "location": [
                  [13, 476],
                  [91, 332],
              "location": [
                  [13, 476],
                  [91, 332],
                  [125, 351],
                  [48, 494]
```

```
        ]
      },
      {
        "words": "Text block 2",
        "confidence": 0.9910,
        "location": [
          [13, 476],
          [91, 332],
          [125, 351],
          [48, 494]
        ]
      }
    ],
    "extracted_data": {}
  }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class RecognizeWebImageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
```

```
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeWebImageRequest request = new RecognizeWebImageRequest();
        WebImageRequestBody body = new WebImageRequestBody();
        List<String> listbodyExtractType = new ArrayList<>();
        listbodyExtractType.add("contact_info");
        listbodyExtractType.add("image_size");
        body.withDetectFont(true);
        body.withExtractType(listbodyExtractType);
        body.withDetectDirection(true);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeWebImageResponse response = client.recognizeWebImage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class RecognizeWebImageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeWebImageRequest request = new RecognizeWebImageRequest();
        WebImageRequestBody body = new WebImageRequestBody();
        List<String> listbodyExtractType = new ArrayList<>();
        listbodyExtractType.add("contact_info");
        listbodyExtractType.add("image_size");
        body.withDetectFont(true);
        body.withExtractType(listbodyExtractType);
        body.withDetectDirection(true);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeWebImageResponse response = client.recognizeWebImage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeWebImageRequest()
        listExtractTypebody = [
            "contact_info",
            "image_size"
        ]
        request.body = WebImageRequestBody(
            detect_font=True,
            extract_type=listExtractTypebody,
            detect_direction=True,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_web_image(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeWebImageRequest()
        listExtractTypebody = [
            "contact_info",
            "image_size"
        ]
        request.body = WebImageRequestBody(
            detect_font=True,
            extract_type=listExtractTypebody,
            detect_direction=True,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_web_image(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeWebImageRequest{}
    var listExtractTypebody = []string{
        "contact_info",
        "image_size",
    }
    detectFontWebImageRequestBody:= true
    detectDirectionWebImageRequestBody:= true
    imageWebImageRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.WebImageRequestBody{
        DetectFont: &detectFontWebImageRequestBody,
        ExtractType: &listExtractTypebody,
        DetectDirection: &detectDirectionWebImageRequestBody,
        Image: &imageWebImageRequestBody,
    }
    response, err := client.RecognizeWebImage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of a web image for recognition. During the recognition, the service verifies the tilt angle of the image, determines the font type to be recognized, and checks whether the image contains contact information.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
```

```
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeWebImageRequest{}
    var listExtractTypebody = []string{
        "contact_info",
        "image_size",
    }
    detectFontWebImageRequestBody:= true
    detectDirectionWebImageRequestBody:= true
    urlWebImageRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.WebImageRequestBody{
        DetectFont: &detectFontWebImageRequestBody,
        ExtractType: &listExtractTypebody,
        DetectDirection: &detectDirectionWebImageRequestBody,
        Url: &urlWebImageRequestBody,
    }
    response, err := client.RecognizeWebImage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

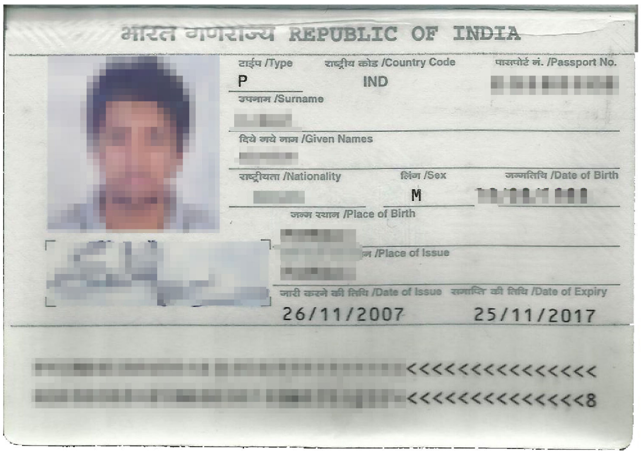
4.5 Passport

Function

This API detects and extracts text from the first page of images of passports and converts the text into a structured JSON format.

In the current version, all fields of a Chinese mainland passport can be recognized. For passports issued by Hong Kong (China), Macao (China), and Taiwan (China), as well as non-Chinese passports, two lines of internationally standardized machine-readable codes on the bottom of each passport can be recognized, and 7 key fields can be extracted from the codes.For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-1 Example passport



भारत गणराज्य REPUBLIC OF INDIA

टाईप /Type: P
राष्ट्रीय कोड /Country Code: IND
पासपोर्ट नं. /Passport No.: [redacted]
उपनाम /Surname: [redacted]
दिये गये नाम /Given Names: [redacted]
राष्ट्रीयता /Nationality: [redacted]
लिंग /Sex: M
जन्मतिथि /Date of Birth: [redacted]
जन्म स्थान /Place of Birth: [redacted]
[redacted] /Place of Issue
जारी करने की तिथि /Date of Issue: 26/11/2007
समाप्ति की तिथि /Date of Expiry: 25/11/2017

[redacted]<<<<<<<<<<<<<<<
[redacted]<<<<<<<<<<<<<<8

Constraints and Limitations

- All fields on Chinese mainland passports can be recognized.
- Passports that are issued by China, Hong Kong (China), Macao (China), Taiwan (China), and other countries and regions and that are with complete machine-readable codes can be recognized.
- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- The information page of the passport to be recognized must occupy more than 25% of the image. When scanning a passport, ensure that the entire page is displayed in the image.
- A passport can be rotated to any angle.
- The passport in the image can be moderately distorted, but the aspect ratio cannot be distorted by more than 10%.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by referring to **Subscribing to an OCR Service**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/passport

Table 4-36 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-37 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-38 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS** **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data. ● The URL cannot contain Chinese characters. If Chinese characters exist, they must be encoded using UTF-8.

Parameter	Mandatory	Type	Description
country_code	No	String	Code of the country where the passport is issued. The passport recognition service is determined based on the country code. - If this parameter is left blank, OCR automatically matches the recognition mode based on the passport type identified by the service. - If you set this parameter to **GENERAL**, the passport is recognized based on the machine-readable code. - If this parameter is set to **CHN**, all fields in the Chinese passport are recognized.

Response Parameters

NOTE

The status code may vary depending on the recognition results. For example, **200** indicates that the API is successfully called, and **400** indicates that the API fails to be called. The following describes the status codes and corresponding response parameters.

Status code: 200

Table 4-39 Response body parameter

Parameter	Type	Description
result	**PassportResult** object	Recognition result This parameter is not returned when the API fails to be called. This parameter consists of the following three parts: 13 key fields, expressed in English; **extra_info**, expressed in local official language; and confidence of key fields. A higher confidence indicates a more accurate result.

Table 4-40 PassportResult

Parameter	Type	Description
passport_type	String	Passport type. The options are as follows: ● **P**: ordinary passport for private affairs ● **W**: diplomatic passport ● **G**: service passport **NOTE** This field is returned only for Chinese mainland passports.
country_code	String	Country code
passport_number	String	Passport number
nationality	String	Nationality of the passport holder **NOTE** This field is returned only for Chinese mainland passports.
surname	String	Family name
given_name	String	Given name
sex	String	Gender
date_of_birth	String	Date of birth, for example, **1990-12-12**
date_of_expiry	String	Passport date of expiry, for example, **2020-07-08**
date_of_issue	String	Date of issue, for example, **2010-07-09** **NOTE** This field is returned only for Chinese mainland passports.
place_of_birth	String	Place of birth **NOTE** This field is returned only for Chinese mainland passports.
place_of_issue	String	Place of issue **NOTE** This field is returned only for Chinese mainland passports.
issuing_authority	String	Issuing authority The abbreviation of the issuing authority of each consulate is not unified. The abbreviation of Chinese issuing authority is **P.R.China**. For example, if the issuing authority is **P.R.C**, the recognition result is **P.R.China**. **NOTE** This field is returned only for Chinese mainland passports.

Parameter	Type	Description
confidence	Object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
extra_info	Object	This parameter is left blank by default. For a Chinese passport, the **extra_info** value contains Chinese character-described fields on the passport, such as the name and place of birth. **NOTE** This field is returned only for Chinese mainland passports.

Status code: 400

Table 4-41 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Passport OCR is deployed in the **CN-Hong Kong** region. The endpoint is **ocr.ap-southeast-1.myhuaweicloud.com or ocr.ap-southeast-1.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/passport**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Authentication**.

- Read the Base64 code of a passport image for recognition.

```
POST https://{endpoint}/v2/{project_id}/ocr/passport
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
    "country_code": "GENERAL"
}
```

- Read the URL of a passport image for recognition.

```
POST https://{endpoint}/v2/{project_id}/ocr/passport
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.xxxx.com/ObjectName",
    "country_code": "GENERAL"
}
```

Example Response

Status code: 200

Chinese passport

```
{
    "result": {
        "passport_type": "P",
        "country_code": "CHN",
        "passport_number": "ED999XXXX",
        "nationality": "CHINESE",
        "surname": "ZHANG",
        "given_name": "SAN",
        "sex": "F",
        "date_of_birth": "1990-12-12",
        "date_of_expiry": "2020-07-08",
        "date_of_issue": "2010-07-09",
        "place_of_birth": "HUNAN",
        "place_of_issue": "GUANGDONG",
        "issuing_authority": "MPS Exit & Entry Administration",
        "extra_info": {
            "local_language": {
                 "name": "Zhang San",
                "sex": "Female",
                "place_of_birth": "Hunan",
                "place_of_issue": "Guangdong",
                "issuing_authority": "xxx Entry and Exit Administration",
                "nationality": "China",
            }
        },
        "confidence": {
            "passport_type": 0.9987,
            "country_code": 0.9897,
            "passport_number": 0.9997,
            "nationality": 0.9977,
            "surname": 0.9729,
            "given_name": 0.9729,
            "sex": 0.9897,
            "date_of_birth": 0.9998,
            "date_of_expiry": 0.9995,
            "date_of_issue": 0.9969,
            "place_of_birth": 0.9937,
            "place_of_issue": 0.9993,
            "issuing_authority": 0.9985
        }
    }
}
```

Non-Chinese passport

```
{
    "result": {
        "country_code": "ETF",
        "surname": "HUZHAO",
        "given_name": "ZHAOMIN DESALEGN ",
        "passport_number": "EP435XXXX",
        "date_of_birth": "1985-09-18",
        "sex": "M",
        "date_of_expiry": "2022-01-15",
        "machine_code": "P<ETFHUZHAO<< ZHAOMIN <DESALEGN<<<<<<<<<<<<<<<",
        "machine_code2": "EP435XXXX7ETF8509185M2201155<<<<<<<<<<<<<<08",
        "extra_info": {},
        "confidence": {
            "country_code": 0.9727,
            "surname": 0.9727,
            "given_name": 0.9727,
            "passport_number": 0.9558,
            "date_of_birth": 0.9558,
            "sex": 0.9558,
            "date_of_expiry": 0.9558
        }
    }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Read the Base64 code of a passport image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizePassportSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizePassportRequest request = new RecognizePassportRequest();
        PassportRequestBody body = new PassportRequestBody();
        body.withCountryCode("CHN");
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAg...");
        request.withBody(body);
        try {
            RecognizePassportResponse response = client.recognizePassport(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Read the URL of a passport image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizePassportSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizePassportRequest request = new RecognizePassportRequest();
        PassportRequestBody body = new PassportRequestBody();
        body.withCountryCode("CHN");
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizePassportResponse response = client.recognizePassport(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Read the Base64 code of a passport image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizePassportRequest()
        request.body = PassportRequestBody(
            country_code="CHN",
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAg..."
        )
        response = client.recognize_passport(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Read the URL of a passport image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizePassportRequest()
        request.body = PassportRequestBody(
            country_code="CHN",
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_passport(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Read the Base64 code of a passport image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizePassportRequest{}
    countryCodePassportRequestBody:= "CHN"
    imagePassportRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAg..."
    request.Body = &model.PassportRequestBody{
        CountryCode: &countryCodePassportRequestBody,
        Image: &imagePassportRequestBody,
    }
    response, err := client.RecognizePassport(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Read the URL of a passport image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizePassportRequest{}
    countryCodePassportRequestBody:= "CHN"
    urlPassportRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.PassportRequestBody{
        CountryCode: &countryCodePassportRequestBody,
        Url: &urlPassportRequestBody,
    }
    response, err := client.RecognizePassport(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

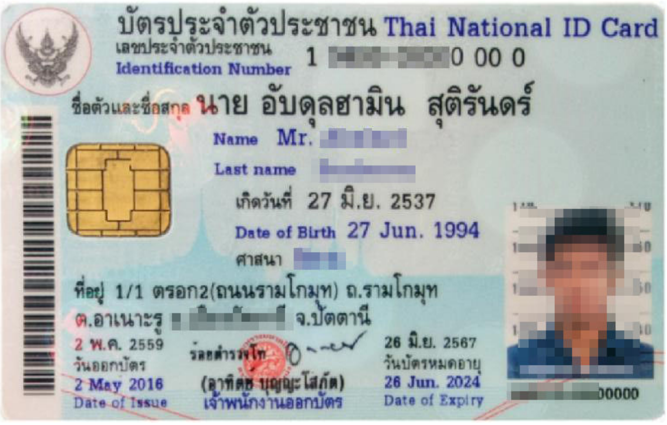
See **Error Codes**.

4.6 Thailand ID Card

Function

This API detects and extracts text from images of Thailand-issued national registration cards and converts the text into a structured JSON format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-2 Example Thailand ID card



บัตรประจำตัวประชาชน Thai National ID Card

เลขประจำตัวประชาชน
Identification Number 1 [redacted]0 00 0

ชื่อตัวและชื่อสกุล นาย อับดุลฮามิน สุติรันดร์

Name Mr. [redacted]

Last name [redacted]

เกิดวันที่ 27 มิ.ย. 2537

Date of Birth 27 Jun. 1994

ศาสนา [redacted]

ที่อยู่ 1/1 ตรอก2(ถนนรามโกมุท) ถ.รามโกมุท
ต.อาเนาะรู [redacted] จ.ปัตตานี

2 พ.ค. 2559
วันออกบัตร
2 May 2016
Date of Issue

ร้อยตำรวจโท
(อาทิตย บุญญะโสภิต)
เจ้าพนักงานออกบัตร

26 มิ.ย. 2567
วันบัตรหมดอายุ
26 Jun. 2024
Date of Expiry

[redacted]00000

Constraints and Limitations

- Only ID cards issued by Thailand can be recognized.
- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card to be recognized must occupy more than 25% of the image. When scanning an ID card, ensure that the entire ID card is displayed in the image.
- An ID card can be rotated to any angle.
- The ID card in the image can be moderately distorted, but the aspect ratio cannot be distorted by more than 10%.

- Illuminated or dark images can be recognized, but the accuracy may be compromised.
- Only the front or back of a single ID card can be identified each time.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/thailand-id-card

Table 4-42 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-43 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.

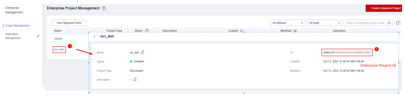
Parameter	Mandatory	Type	Description
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-44 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.
side	No	String	Front or back of the ID card. The options are as follows: ● **front**: front of an ID card ● **back**: back of an ID card If the value of this parameter is empty or not included, the system will automatically recognize whether the image is the front or back of an ID card. It is recommended to set this parameter for higher accuracy.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the Base64 code of the portrait on the ID card image. The options are as follows: ● **true**: The Base64 code of the portrait on the ID card will be returned. ● **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.
return_portrait_location	No	Boolean	Whether to return the position coordinates of the portrait on the ID card image. The options are as follows: ● **true**: The location of the portrait on the ID card will be returned. ● **false**: The location of the portrait on the ID card will not be returned.
return_idcard_type	No	Boolean	Whether to return the ID card type. The options are as follows: ● **true**: The ID card type will be returned, indicating that the ID card is the original ID card or copy of the original ID card. ● **false**: The ID card type will not be returned.
return_text_location	No	Boolean	Location of a text block. The options are as follows: **true**: All text blocks will be returned. **false**: Text blocks will not be returned. If this parameter is not specified, the system does not return the location of any text blocks by default. If a non-Boolean value is entered, an error message will be displayed, indicating that the parameter is invalid.

Response Parameters

Status code: 200

Table 4-45 Response body parameter

Parameter	Type	Description
result	**ThailandIdcardResult** object	Recognition result This parameter is not returned when the API fails to be called.

Table 4-46 ThailandIdcardResult

Parameter	Type	Description
type	String	ID card type. The value **normal** indicates a regular Thailand ID card, and the value **pink** indicates a foreigner's ID card.
name_en	String	English name
ref_number	String	Reference number
side	String	Front or back of an ID card. Value options are **front** and **back**.
id_number	String	ID number
name_th	String	Thai name
first_name_en	String	Name, in English
last_name_en	String	English surname
date_of_birth_th	String	Date of birth, in Thai
date_of_birth_en	String	Date of birth, in English
religion_th	String	Religion
address_th	String	Address
date_of_issue_th	String	Date of issue, in Thai
date_of_issue_en	String	Date of issue, in English
date_of_expiry_th	String	Date of expiry, in Thai
date_of_expiry_en	String	Date of expiry, in English
serial_number	String	Serial number
card_number	String	ID number on the back of the ID card

Parameter	Type	Description
laser_number	String	Laser code
confidence	**ThailandIdcardConfidence** object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
portrait_image	String	Base64 code of the portrait. This parameter is returned only when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
idcard_type	String	ID card type. This parameter is returned only when **return_idcard_type** is set to **true**. The options are as follows: ● **normal**: original ID card ● **copy**: copy of the ID card
text_location	Object	Location of all fields identified on the original image, including the two-dimensional coordinates (x,y) of the four vertices in all text areas. The image coordinate system is used. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.

Table 4-47 ThailandIdcardConfidence

Parameter	Type	Description
id_number	Float	Confidence of the ID number
name_th	Float	Confidence of the Thai name
name_en	Float	Confidence of the English name
ref_number	Float	Confidence of the reference number
first_name_en	Float	Confidence of the English given name

Parameter	Type	Description
last_name_en	Float	Confidence of the English family name
date_of_birth_th	Float	Confidence of the birth date in Thai An example of the return value is **5 ก. พ. 2493**.
date_of_birth_en	Float	Confidence of the birth date in English An example of the return value is **5 Feb. 1950**.
religion_th	Float	Confidence of the religion
address_th	Float	Confidence of the address
date_of_issue_th	Float	Confidence of the issuance date in Thai
date_of_issue_en	Float	Confidence of the issuance date in English
date_of_expiry_th	Float	Confidence of the validity period in Thai
date_of_expiry_en	Float	Confidence of the validity period in English
serial_number	Float	Confidence of the serial number
card_number	Float	Confidence of the card number on the back of the ID card
laser_number	Float	Confidence of the laser code

Status code: 200

Table 4-48 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Thailand ID Card OCR is deployed in the **AP-Bangkok** region. The endpoint is **ocr.ap-southeast-2.myhuaweicloud.com** or **ocr.ap-southeast-2.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/thailand-id-card**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the image Base64 string.)

```
POST  https://{endpoint}/v2/{project_id}/ocr/thailand-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:

{
  "image" : "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "side" : "front",
  "return_portrait_image" : true,
  "return_portrait_location" : true,
  "return_idcard_type" : true
}
```

- Request example (Method 2: Use the image URL.)

```
POST  https://{endpoint}/v2/{project_id}/ocr/thailand-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.xxxx.com/ObjectName"
}
```

- Sample code for a Python 3 request (For codes in other languages, refer to the following sample or use OCR SDK.)

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v2/{project_id}/ocr/thailand-id-card"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/thailand-id-card-demo.png' # Read a local image.
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use the Base64 code of the
image.
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

Example Response

Status code: 200

Example response for a successful request (the front)

```
{
  "result" : {
    "side" : "front",
    "id_number" : "X XXXX XXXXX XX X",
    "name_th" : "XXX",
    "first_name_en" : "XX",
    "last_name_en" : "XX",
    "date_of_birth_th" : "5 ก.พ. 2493",
    "date_of_birth_en" : "5 Feb. 1950",
    "religion_th" : "XX",
    "address_th" : "XXXXX",
    "date_of_issue_th" : "XX",
    "date_of_issue_en" : "4 Mar. 2011",
    "date_of_expiry_th" : "22 ก.พ. 2561",
    "date_of_expiry_en" : "22 Feb. 2018",
    "serial_number" : "XXXX-XX-XXXXX",
    "confidence" : {
      "id_number" : 0.9999,
      "name_th" : 0.9994,
      "first_name_en" : 0.998,
      "last_name_en" : 0.9997,
      "date_of_birth_th" : 0.9996,
      "date_of_birth_en" : 0.9997,
      "religion_th" : 0.686,
      "address_th" : 0.624,
      "date_of_issue_th" : 1,
      "date_of_issue_en" : 1,
      "date_of_expiry_th" : 0.9969,
      "date_of_expiry_en" : 0.61,
      "serial_number" : 0.9887
    },
    "portrait_image" : "/9j/4AA... ",
    "portrait_location" : [ [ 576, 237 ], [ 741, 237 ], [ 739, 430 ], [ 574, 431 ] ],
    "idcard_type" : "normal"
  }
}
```

Example response for a successful request (the back)

```
{
    "result": {
        "side": "back",
        "card_number": "XXXX-XXX-XX",
        "laser_number": "XXXX-XXXXXXX-XX",
        "confidence": {
            "id_number": 0.9999,
            "laser_number": 0.9994
        }
    }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Thailand ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeThailandIdcardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeThailandIdcardRequest request = new RecognizeThailandIdcardRequest();
        ThailandIdcardRequestBody body = new ThailandIdcardRequestBody();
        body.withReturnIdcardType(true);
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withSide("front");
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeThailandIdcardResponse response = client.recognizeThailandIdcard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Thailand ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;
```

```
public class RecognizeThailandIdcardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeThailandIdcardRequest request = new RecognizeThailandIdcardRequest();
        ThailandIdcardRequestBody body = new ThailandIdcardRequestBody();
        body.withReturnIdcardType(true);
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withSide("front");
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeThailandIdcardResponse response = client.recognizeThailandIdcard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Thailand ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeThailandIdcardRequest()
        request.body = ThailandIdcardRequestBody(
            return_idcard_type=True,
            return_portrait_location=True,
            return_portrait_image=True,
            side="front",
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_thailand_idcard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Thailand ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeThailandIdcardRequest()
        request.body = ThailandIdcardRequestBody(
            return_idcard_type=True,
            return_portrait_location=True,
            return_portrait_image=True,
            side="front",
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_thailand_idcard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Thailand ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeThailandIdcardRequest{}
    returnIdcardTypeThailandIdcardRequestBody:= true
    returnPortraitLocationThailandIdcardRequestBody:= true
    returnPortraitImageThailandIdcardRequestBody:= true
    sideThailandIdcardRequestBody:= "front"
    imageThailandIdcardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.ThailandIdcardRequestBody{
        ReturnIdcardType: &returnIdcardTypeThailandIdcardRequestBody,
        ReturnPortraitLocation: &returnPortraitLocationThailandIdcardRequestBody,
        ReturnPortraitImage: &returnPortraitImageThailandIdcardRequestBody,
        Side: &sideThailandIdcardRequestBody,
        Image: &imageThailandIdcardRequestBody,
    }
    response, err := client.RecognizeThailandIdcard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Thailand ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeThailandIdcardRequest{}
    returnIdcardTypeThailandIdcardRequestBody:= true
    returnPortraitLocationThailandIdcardRequestBody:= true
    returnPortraitImageThailandIdcardRequestBody:= true
    sideThailandIdcardRequestBody:= "front"
    urlThailandIdcardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.ThailandIdcardRequestBody{
        ReturnIdcardType: &returnIdcardTypeThailandIdcardRequestBody,
        ReturnPortraitLocation: &returnPortraitLocationThailandIdcardRequestBody,
        ReturnPortraitImage: &returnPortraitImageThailandIdcardRequestBody,
        Side: &sideThailandIdcardRequestBody,
        Url: &urlThailandIdcardRequestBody,
    }
    response, err := client.RecognizeThailandIdcard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.7 Cambodian ID Card

Function

This API detects and extracts text from images of Cambodia-issued ID cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Currently, only the front of an ID card can be recognized each time.
- Only images in PNG, JPG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card can be rotated to any angle.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/cambodian-idcard

Table 4-49 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-50 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

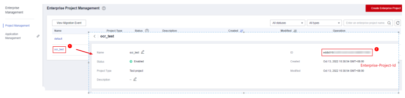
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-51 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 code of an image. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. URL of the image file. Currently, URLs for anonymously and publicly authorized or external access to image files stored on OBS are supported.
return_portrait_image	No	Boolean	Whether to return the Base64 code of the portrait on the ID card image. The options are as follows: ● **true**: The Base64 code of the portrait on the ID card will be returned. ● **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.
return_portrait_location	No	Boolean	Whether to return the position coordinates of the portrait on the ID card image. The options are as follows: ● **true**: The location of the portrait on the ID card will be returned. ● **false**: The location of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the location of the portrait on the ID card will not be returned.

Parameter	Mandatory	Type	Description
return_idcard_type	No	Boolean	Whether to return the ID card type. The options are as follows: - **true**: The ID card type will be returned, indicating that the ID card is the original ID card or copy of the original ID card. - **false**: The ID card type will not be returned.

Response Parameters

Status code: 200

Table 4-52 Response body parameter

Parameter	Type	Description
result	**CambodianIdCardResult** object	Recognition result This parameter is not included when the API fails to be called.

Table 4-53 CambodianIdCardResult

Parameter	Type	Description
id_number	String	ID number
name_kh	String	Khmer name
name_en	String	Name in English
birth_date	String	Date of birth
sex	String	Gender
height	String	Height
birth_place	String	Place of birth
address	String	Addresses, separated by spaces
issue_date	String	Date of issue
expiry_date	String	Date of expiry
description	String	Personal features in the image

Parameter	Type	Description
machine_code1	String	Machine code in the first line
machine_code2	String	Machine code in the second line
machine_code3	String	Machine code in the third line
portrait_image	String	Base64 code of the portrait. This parameter is available only when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
idcard_type	String	ID card type. This parameter is returned only when **idcard_type** is set to **true**. The options are as follows: ● **normal**: original ID card ● **copy**: copy of the ID card
confidence	Object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Status code: 400

Table 4-54 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

- Request example (Method 1: Use the image Base64 string.)

```
POST https://{endpoint}/v2/{project_id}/ocr/cambodian-idcard
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
 }
```

- Request example (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/cambodian-idcard
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "url":"https://BucketName.obs.myhwclouds.com/ObjectName"
 }
```

Example Response

Status code: 200

Example response for a successful request

```
{
"result": {
        "id_number": "XXXXX",
        "name_kh": "យឹXXXXX",
        "name_en": "YOENG THXXXXX",
        "birth_date": "១០.០៧.១៩៩៩",
        "sex": "ស្រី",
        "height": "១៥៨ ស.ម",
        "birth_place": "XXXXX",
        "address": " XXXXX",
        "issue_date": "១៥.១២.២០១៥",
        "expiry_date": "១៤.១២.២០២៥",
        "description": "XXXXXX",
        "machine_code1": "IDKHM040XXXXXXX<<<<<<<<<<<<<<<",
        "machine_code2": "990XXX2F25XXXXXXX<<<<<<<<<<<2",
        "machine_code3": "YOENG<<THXXXXX<<<<<<<<<<<<<<<",
         "idcard_type": "normal",
        "confidence": {
            "id_number": 0.8345,
            "name_kh": 0.8721,
            "name_en": 0.7191,
            "birth_date": 0.7749,
            "sex": 0.8216,
            "height": 0.6493,
            "birth_place": 0.8282,
            "address": 0.91185,
            "issue_date": 0.7509,
            "expiry_date": 0.7983,
            "description": 0.9091,
            "machine_code1": 0.9047,
            "machine_code2": 0.9292,
            "machine_code3": 0.898,
            "idcard_type": 0.998
        },
        "portrait_image": "/9j/4AA....",
        "portrait_location": [
            [
                53,
                341
            ],
            [
                185,
                340
            ],
            [
                179,
                504
            ],
            [
                42,
                505
            ]
        ]
    }
    }
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Cambodian ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;
```

```
public class RecognizeCambodianIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeCambodianIdCardRequest request = new RecognizeCambodianIdCardRequest();
        CambodianIdCardRequestBody body = new CambodianIdCardRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeCambodianIdCardResponse response = client.recognizeCambodianIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Cambodian ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeCambodianIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeCambodianIdCardRequest request = new RecognizeCambodianIdCardRequest();
        CambodianIdCardRequestBody body = new CambodianIdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeCambodianIdCardResponse response = client.recognizeCambodianIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Cambodian ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeCambodianIdCardRequest()
        request.body = CambodianIdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_cambodian_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Cambodian ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeCambodianIdCardRequest()
        request.body = CambodianIdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_cambodian_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Cambodian ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeCambodianIdCardRequest{}
    imageCambodianIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.CambodianIdCardRequestBody{
        Image: &imageCambodianIdCardRequestBody,
    }
    response, err := client.RecognizeCambodianIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Cambodian ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeCambodianIdCardRequest{}
    urlCambodianIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.CambodianIdCardRequestBody{
        Url: &urlCambodianIdCardRequestBody,
    }
    response, err := client.RecognizeCambodianIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

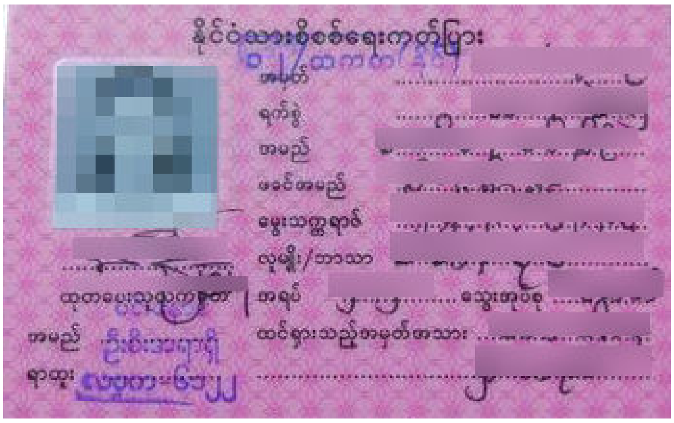
See **Error Codes**.

4.8 Myanmar ID Card

Function

This API detects and extracts text from Myanmar-issued national registration cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-3 Example Myanmar ID card



နိုင်ငံသားစိစစ်ရေးကတ်ပြား

အမှတ် [redacted]

ရက်စွဲ [redacted]

အမည် [redacted]

ဖခင်အမည် [redacted]

မွေးသက္ကရာဇ် [redacted]

လူမျိုး/ဘာသာ [redacted]

[redacted]

အရပ် [redacted] သွေးအုပ်စု [redacted]

ထုတ်ပေးသူအရာရှိ

ထင်ရှားသည့်အမှတ်အသား [redacted]

အမည်

ရာထူး

[redacted]

Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,000 pixels.
- An ID card to be recognized must occupy more than 25% of the image. When scanning an ID card, ensure that the entire ID card is displayed in the image.
- An ID card can be rotated to any angle.
- The ID card in the image can be moderately distorted, but the aspect ratio cannot be distorted by more than 10%.

- Illuminated or dark images can be recognized, but the accuracy may be compromised.
- Only the front or back of a single ID card can be identified each time.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

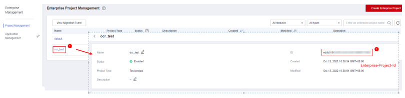
Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/myanmar-id-card

Table 4-55 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Parameter	Mandatory	Description
Enterprise-Project-Id	No	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 Enterprise-Project-Id For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Request Parameters

Table 4-56 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-57 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,000 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.
convert_unicode	No	Boolean	Output format. The options are as follows: ● **true**: The output will be in the Unicode format. ● **false**: The output will be in the zawgyi format. If this parameter is not specified or does not exist, the output will be in the zawgyi format by default.

Parameter	Mandatory	Type	Description
return_confidence	No	Boolean	Whether to return the confidence. The options are as follows: **true**: The confidence will be returned. **false**: The confidence will not be returned. If this parameter is not specified, the system does not return the confidence by default. If a non-Boolean value is entered, an error message will be displayed, indicating that the parameter is invalid.
return_portrait_image	No	Boolean	Whether to return the Base64 code of the portrait on the ID card image. The options are as follows: ● **true**: The Base64 code of the portrait on the ID card will be returned. ● **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the ID card image. The options are as follows: ● **true**: The location of the portrait on the ID card will be returned. ● **false**: The location of the portrait on the ID card will not be returned.
return_idcard_type	No	Boolean	Whether to return the ID card type. The options are as follows: ● **true**: The ID card type, either original or copy of the original will be returned. ● **false**: The ID card type will not be returned.
return_translation	No	Boolean	Whether to return translation information. The options are as follows: ● **true**: The translation information is returned. ● **false**: The translation information is not returned.

Response Parameters

Status code: 200

Table 4-58 Response body parameter

Parameter	Type	Description
result	**MyanmarIdcardResult** object	Calling result This parameter is not returned when the API fails to be called.

Table 4-59 MyanmarIdcardResult

Parameter	Type	Description
side	String	Front or back of a national registration card. The options are **front** and **back**.
class	String	ID card type. The options are as follows: ● **new_version**: new version of ID card ● **old_version**: old version of ID card
nrc_id	String	ID number
issue_date	String	Date of issue
name	String	Name
father_name	String	Father's name
birth	String	Date of birth
bloodlines_religion	String	Ethnic group or religion
height	String	Height
blood_group	String	Blood type
card_id	String	Card number on the back of the ID card
nrc_id_back	String	ID number on the back of the ID card
profession	String	Occupation
address	String	Address
confidence	**MyanmarIdcardConfidence** object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Parameter	Type	Description
portrait_image	String	Base64 code of the portrait. This parameter is returned only when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
idcard_type	String	ID card type. This parameter is returned only when **return_idcard_type** is set to **true**. The options are as follows: ● **normal**: original ID card ● **copy**: copy of the ID card
translation_info	**MyanmarIdcardTranslationInfo** object	Translation information. This parameter is returned only when **return_translation** is set to **true**. This field contains information about **name_translation** and **nrc_id_translation**. When no appropriate translation fields can be found, the original Burmese characters are retained.

Table 4-60 MyanmarIdcardConfidence

Parameter	Type	Description
nrc_id	Float	Confidence of the ID number
issue_date	Float	Confidence of the issuance date
name	Float	Confidence of the name
birth	Float	Confidence of the birth date
bloodlines_religion	Float	Confidence of the ethnic group or religion
height	Float	Confidence of the height
blood_group	Float	Confidence of the blood type
card_id	Float	Confidence of the card number on the back of the ID card
nrc_id_back	Float	ID number on the back of the ID card

Parameter	Type	Description
profession	Float	Confidence of the occupation
address	Float	Confidence of the address

Table 4-61 MyanmarIdcardTranslationInfo

Parameter	Type	Description
name_translation	String	Name translation. This parameter is returned only when **return_translation** is set to **true**.
father_name_translation	String	Translation of the father's name. This parameter is returned only when **return_translation** is set to **true**.
nrc_id_translation	String	Translation of the ID number. This parameter is returned only when **return_translation** is set to **true**.
birth_translation	String	Birth date translation. This parameter is returned only when **return_translation** is set to **true**.

Status code: 400

Table 4-62 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Myanmar ID Card OCR is deployed in the **CN-Hong Kong** region. The endpoint is **ocr.ap-southeast-1.myhuaweicloud.com** or **ocr.ap-southeast-1.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/myanmar-id-card**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the Base64 encoded string of an image.)

```
POST https://{endpoint}/v2/{project_id}/ocr/myanmar-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAAgABwESAAd...",
    "convert_unicode": true,
    "return_confidence": true,
    "return_portrait_image": true,
    "return_portrait_location": true,
    "return_idcard_type": true
}
```

- Request example (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/myanmar-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.xxxx.com/ObjectName",
    "convert_unicode": true,
    "return_confidence": true,
    "return_portrait_image": true,
    "return_portrait_location": true,
    "return_idcard_type": true
}
```

- Sample code for a Python 3 request (For codes in other languages, refer to the following sample or use OCR SDK.)

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v2/{project_id}/ocr/myanmar-id-card"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/myanmar-id-card-demo.png'  # Read a local image.
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use the Base64 code of the image.
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

Example Response

Status code: 200

Example response for a successful request (the front)

```
{
  "result" : {
    "side" : "front",
    "nrc_id" : "XXXXXXXX",
    "issue_date" : "□□.□.□□□□",
    "name" : "XXXX",
    "father_name" : "XXX",
    "birth" : "□□.□□.□□□□",
    "bloodlines_religion" : "□□□□□□□□□",
    "height" : "□'□\"",
    "blood_group" : "□□",
    "class" : "new_version",
    "confidence" : {
      "nrc_id" : 0.7514,
      "issue_date" : 0.5385,
      "name" : 0.6641,
      "birth" : 0.5216,
      "bloodlines_religion" : 0.9774,
      "height" : 0.7526,
      "blood_group" : 0.7541
    },
    "portrait_image" : "/9j/4AA... ",
    "portrait_location" : [ [ 106, 178 ], [ 369, 181 ], [ 366, 448 ], [ 108, 445 ] ],
    "idcard_type" : "normal",
    "translation_info" : {
      "name_translation" : "Ma Sandar Phy",
      "father_name_translation" : "U Thein Po",
      "birth_translation" : "7.5.1992",
      "nrc_id_translation" : "9/MaKhaNa(C)251959"
    }
  }
}
```

Successful response example (the back)

```
{
    "result":{
        "side": "back",
        "card_id": "XXXXXXX",
        "nrc_id_back": "",
        "profession": "□□□□□□□□□□",
        "address": "XXXXXXXX",
        "class": "new_version",
        "confidence": {
            "card_id": 0.9878,
            "nrc_id_back": 0.9595,
            "profession": 0.9995,
            "address": 0.9299
        },
        "idcard_type": "normal"
    }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Myanmar ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMyanmarIdcardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMyanmarIdcardRequest request = new RecognizeMyanmarIdcardRequest();
        MyanmarIdcardRequestBody body = new MyanmarIdcardRequestBody();
        body.withReturnIdcardType(true);
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withReturnConfidence(true);
        body.withConvertUnicode(true);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeMyanmarIdcardResponse response = client.recognizeMyanmarIdcard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Myanmar ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMyanmarIdcardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMyanmarIdcardRequest request = new RecognizeMyanmarIdcardRequest();
        MyanmarIdcardRequestBody body = new MyanmarIdcardRequestBody();
        body.withReturnIdcardType(true);
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withReturnConfidence(true);
        body.withConvertUnicode(true);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeMyanmarIdcardResponse response = client.recognizeMyanmarIdcard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Myanmar ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMyanmarIdcardRequest()
        request.body = MyanmarIdcardRequestBody(
            return_idcard_type=True,
            return_portrait_location=True,
            return_portrait_image=True,
            return_confidence=True,
            convert_unicode=True,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_myanmar_idcard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Myanmar ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMyanmarIdcardRequest()
        request.body = MyanmarIdcardRequestBody(
            return_idcard_type=True,
            return_portrait_location=True,
            return_portrait_image=True,
            return_confidence=True,
            convert_unicode=True,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_myanmar_idcard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Myanmar ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMyanmarIdcardRequest{}
    returnIdcardTypeMyanmarIdcardRequestBody:= true
    returnPortraitLocationMyanmarIdcardRequestBody:= true
    returnPortraitImageMyanmarIdcardRequestBody:= true
    returnConfidenceMyanmarIdcardRequestBody:= true
    convertUnicodeMyanmarIdcardRequestBody:= true
    imageMyanmarIdcardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.MyanmarIdcardRequestBody{
        ReturnIdcardType: &returnIdcardTypeMyanmarIdcardRequestBody,
        ReturnPortraitLocation: &returnPortraitLocationMyanmarIdcardRequestBody,
        ReturnPortraitImage: &returnPortraitImageMyanmarIdcardRequestBody,
        ReturnConfidence: &returnConfidenceMyanmarIdcardRequestBody,
        ConvertUnicode: &convertUnicodeMyanmarIdcardRequestBody,
        Image: &imageMyanmarIdcardRequestBody,
    }
    response, err := client.RecognizeMyanmarIdcard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Myanmar ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMyanmarIdcardRequest{}
    returnIdcardTypeMyanmarIdcardRequestBody:= true
    returnPortraitLocationMyanmarIdcardRequestBody:= true
    returnPortraitImageMyanmarIdcardRequestBody:= true
    returnConfidenceMyanmarIdcardRequestBody:= true
    convertUnicodeMyanmarIdcardRequestBody:= true
    urlMyanmarIdcardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.MyanmarIdcardRequestBody{
        ReturnIdcardType: &returnIdcardTypeMyanmarIdcardRequestBody,
        ReturnPortraitLocation: &returnPortraitLocationMyanmarIdcardRequestBody,
        ReturnPortraitImage: &returnPortraitImageMyanmarIdcardRequestBody,
        ReturnConfidence: &returnConfidenceMyanmarIdcardRequestBody,
        ConvertUnicode: &convertUnicodeMyanmarIdcardRequestBody,
        Url: &urlMyanmarIdcardRequestBody,
    }
    response, err := client.RecognizeMyanmarIdcard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

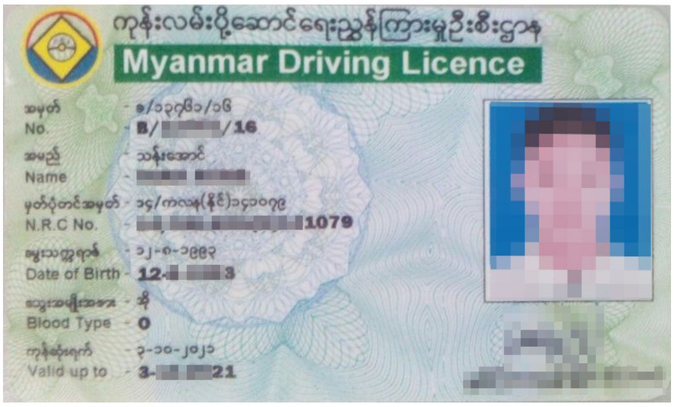
See **Error Codes**.

4.9 Myanmar Driving License

Function

This API detects and extracts text from images of Myanmar-issued driver's licenses and converts the text into a structured JSON format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-4 Example Myanmar driving license



ကုန်းလမ်းပို့ဆောင်ရေးညွှန်ကြားမှုဦးစီးဌာန
Myanmar Driving Licence

အမှတ် - ဗ/၁၃၇၆၁/၁၆
No. - B/[redacted]/16
အမည် - သန်းအောင်
Name - [redacted]
မှတ်ပုံတင်အမှတ် - ၁၄/ကလန(နိုင်)၁၄၁၀၇၉
N.R.C No. - [redacted]1079
မွေးသက္ကရာဇ် - ၁၂-၈-၁၉၉၃
Date of Birth - 12-[redacted]3
သွေးအမျိုးအစား - အို
Blood Type - O
ကုန်ဆုံးရက် - ၃-၁၀-၂၀၂၁
Valid up to - 3-[redacted]21

[redacted]

[redacted]

Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 4,096 pixels.
- Currently, only the front of a driving license can be recognized each time.
- A driving license can be rotated to any angle.
- The driving license in the image can be moderately distorted, but the aspect ratio cannot be distorted by more than 10%.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/myanmar-driver-license

Table 4-63 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-64 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

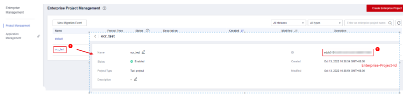
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-65 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 4,096 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.
convert_unicode	No	Boolean	Output format. The options are as follows: ● **true**: The output will be in the Unicode format. ● **false**: The output will be in the zawgyi format. If this parameter is not specified or does not exist, the output will be in the zawgyi format by default.

Response Parameters

Status code: 200

Table 4-66 Response body parameter

Parameter	Type	Description
result	**MyanmarDriverLicenseResult** object	Calling result This parameter is not returned when the API fails to be called.

Table 4-67 MyanmarDriverLicenseResult

Parameter	Type	Description
card_number	String	Number of the Myanmar driving license, in Burmese
card_number_en	String	Number of the Myanmar driving license, in English
name	String	Name, in Burmese
name_en	String	Name, in English
nrc_id	String	National registration card number, in Burmese
nrc_id_en	String	National registration card number, in English
birth	String	Date of birth, in Burmese
birth_en	String	Date of birth, in English
blood_group	String	Blood type, in Burmese
blood_group_en	String	Blood type, in English
expiried_date	String	Date of expiry, in Burmese
expiried_date_en	String	Date of expiry, in English
confidence	**MyanmarDriverLicenseConfidence** object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Table 4-68 MyanmarDriverLicenseConfidence

Parameter	Type	Description
card_number	Float	Confidence of the driving license number in Burmese

Parameter	Type	Description
card_number_en	Float	Confidence of the driving license number in English
name	Float	Confidence of the Myanmar name
name_en	Float	Confidence of the English given name
nrc_id	Float	Confidence of the NRC number in Burmese
nrc_id_en	Float	Confidence of the NRC number in English
birth	Float	Confidence of the birth date in Burmese
birth_en	Float	Confidence of the birth date in English
blood_group	Float	Confidence of the blood type in Burmese
blood_group_en	Float	Confidence of the blood type in English
expiried_date	Float	Confidence of the validity period in Burmese
expiried_date_en	Float	Confidence of the validity period in English

Status code: 400

Table 4-69 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Myanmar Driving License OCR is deployed in the **CN-Hong Kong** region. The endpoint is **ocr.ap-southeast-1.myhuaweicloud.com** or **ocr.ap-southeast-1.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/myanmar-driver-license**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the image Base64 string.)

```
POST https://{endpoint}/v2/{project_id}/ocr/myanmar-driver-license

Request Header:
Content-Type: application/json
X-Auth-Token: 
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
  "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "convert_unicode": true
}
```

- Request example (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/myanmar-driver-license

Request Header:
Content-Type: application/json
X-Auth-Token: 
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
  "url":"https://BucketName.obs.xxxx.com/ObjectName",
  "convert_unicode": true
}
```

- Sample code for a Python 3 request (For codes in other languages, refer to the following sample or use OCR SDK.)

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v2/{project_id}/ocr/myanmar-driver-license"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/myanmar-driver-license-demo.png' # Read a local image.
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use the Base64 code of the 
image.
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "card_number" : "XXXX",
    "card_number_en" : "XXXX",
    "name" : "XXXX",
    "name_en" : "XXXX",
    "nrc_id" : "XXX",
    "nrc_id_en" : "XXX",
    "birth" : "□-□-□□□□",
    "birth_en" : "5-5-1992",
    "blood_group" : "□□",
    "blood_group_en" : "2",
    "expiration_date" : "□□-□-□□□□",
    "expiration_date_en" : "13-1-2020",
    "confidence" : {
      "card_number" : 0.8252,
      "card_number_en" : 0.8971,
      "name" : 0.985,
      "name_en" : 0.9528,
      "nrc_id" : 0.9972,
      "nrc_id_en" : 0.9993,
      "birth" : 0.9998,
      "birth_en" : 0.8973,
      "blood_group" : 0.6772,
      "blood_group_en" : 0.6721,
      "expiration_date" : 0.9994,
      "expiration_date_en" : 0.758
    }
  }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Myanmar driving license image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMyanmarDriverLicenseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMyanmarDriverLicenseRequest request = new
RecognizeMyanmarDriverLicenseRequest();
        MyanmarDriverLicenseRequestBody body = new MyanmarDriverLicenseRequestBody();
        body.withConvertUnicode(true);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeMyanmarDriverLicenseResponse response =
client.recognizeMyanmarDriverLicense(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Myanmar driving license image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMyanmarDriverLicenseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMyanmarDriverLicenseRequest request = new 
RecognizeMyanmarDriverLicenseRequest();
        MyanmarDriverLicenseRequestBody body = new MyanmarDriverLicenseRequestBody();
        body.withConvertUnicode(true);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeMyanmarDriverLicenseResponse response = 
client.recognizeMyanmarDriverLicense(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Myanmar driving license image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMyanmarDriverLicenseRequest()
        request.body = MyanmarDriverLicenseRequestBody(
            convert_unicode=True,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_myanmar_driver_license(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Myanmar driving license image for recognition.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMyanmarDriverLicenseRequest()
        request.body = MyanmarDriverLicenseRequestBody(
            convert_unicode=True,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_myanmar_driver_license(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Myanmar driving license image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMyanmarDriverLicenseRequest{}
    convertUnicodeMyanmarDriverLicenseRequestBody:= true
    imageMyanmarDriverLicenseRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.MyanmarDriverLicenseRequestBody{
        ConvertUnicode: &convertUnicodeMyanmarDriverLicenseRequestBody,
        Image: &imageMyanmarDriverLicenseRequestBody,
    }
    response, err := client.RecognizeMyanmarDriverLicense(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Myanmar driving license image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMyanmarDriverLicenseRequest{}
    convertUnicodeMyanmarDriverLicenseRequestBody:= true
    urlMyanmarDriverLicenseRequestBody:= "https://BucketName.obs.myhuaweicloud.com/
ObjectName"
    request.Body = &model.MyanmarDriverLicenseRequestBody{
        ConvertUnicode: &convertUnicodeMyanmarDriverLicenseRequestBody,
        Url: &urlMyanmarDriverLicenseRequestBody,
    }
    response, err := client.RecognizeMyanmarDriverLicense(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

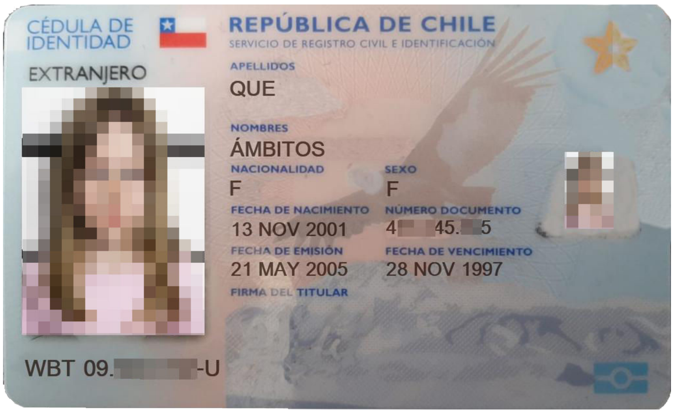
See **Error Codes**.

4.10 Chile ID Card

Function

This API detects and extracts text from images of Chile-issued ID cards and converts the text into JSON format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-5 Example Chile ID card



Constraints and Limitations

- Only images in PNG, JPG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.

- Currently, only the front of an ID card can be recognized each time.
- An ID card can be rotated to any angle.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before using the service for the first time, you need to enable the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/chile-id-card

Table 4-70 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**. The endpoint of the Chile ID Card OCR API is **ocr.la-south-2.myhuaweicloud.com**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-71 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

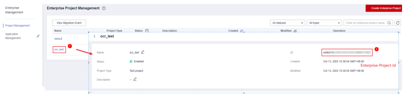
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR allows you to use Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016 Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: • If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. • If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. • If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-72 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Response Parameters

Status code: 200

Table 4-73 Response body parameter

Parameter	Type	Description
result	**ChileIdCardResult** object	Calling result This parameter is not returned when the API fails to be called.

Table 4-74 ChileIdCardResult

Parameter	Type	Description
surname	Array of strings	Last name
given_name	String	First name
nationality	String	Nationality
sex	String	Gender
birth	String	Date of birth
issue_date	String	Date of issue
expiry_date	String	Date of expiry
document_number	String	Document number
number	String	ID number
confidence	**ChileIdCardConfidence** object	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Table 4-75 ChileIdCardConfidence

Parameter	Type	Description
surname	Float	Confidence of the last name
given_name	Float	Confidence of the given name
nationality	Float	Confidence of the nationality
sex	Float	Confidence of the gender
birth	Float	Confidence of the birth date
issue_date	Float	Confidence of the issuance date
expiry_date	Float	Confidence of the validity period
document_number	Float	Confidence of the document number
number	Float	Confidence of the ID number

Status code: 400

Table 4-76 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Chile ID Card OCR is deployed in the **LA-Santiago** region. The endpoint is **ocr.la-south-2.myhuaweicloud.com** or **ocr.la-south-2.myhuaweicloud.cn**. The request URL is **https://ocr.la-south-2.myhuaweicloud.com/v2/{project_id}/ocr/chile-id-card**. **project_id** indicates the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the Base64 encoded string of an image.)
```
POST https://ocr.la-south-2.myhuaweicloud.com/v2/{project_id}/ocr/chile-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgA..."
}
```
- Request example (Method 2: Use the image URL.)
```
POST https://ocr.la-south-2.myhuaweicloud.com/v2/{project_id}/ocr/chile-id-card

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "url":"https://BucketName.obs.xxxx.com/ObjectName"
}
```
- Sample code for a Python 3 request (For codes in other languages, refer to the following sample or use OCR SDK.)
```
# encoding:utf-8

import requests
import base64
```

```
url = "https://ocr.la-south-2.myhuaweicloud.com/v2/{project_id}/ocr/chile-id-card"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/chile-id-card-demo.png' # Read a local image.
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use the Base64 code of the image.
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result": {
      "surname": [
          "FERNANDEZ",
          "GATICA"
      ],
      "given_name": "MARCELA CAROLINA",
      "nationality": "CHILENA",
      "sex": "F",
      "birth": "21 FEB 1982",
      "document_number": "100000001",
      "issue_date": "1 SEP 2013",
      "expiry_date": "10 AGO 2023",
      "number": "12.749.625-K",
      "confidence": {
          "surname": 0.9584,
          "given_name": 0.8106,
          "nationality": 0.7026,
          "sex": 0.5879,
          "birth": 0.9305,
          "document_number": 0.8181,
          "issue_date": 0.8518,
          "expiry_date": 0.7757,
          "number": 0.9528
      }
  }
}
```

Status code: 400

Example response for a failed request

```
{
   "error_code": "AIS.0103",
   "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Chile ID card image for recognition.

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeChileIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeChileIdCardRequest request = new RecognizeChileIdCardRequest();
        ChileIdCardRequestBody body = new ChileIdCardRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeChileIdCardResponse response = client.recognizeChileIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Chile ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeChileIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeChileIdCardRequest request = new RecognizeChileIdCardRequest();
        ChileIdCardRequestBody body = new ChileIdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeChileIdCardResponse response = client.recognizeChileIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Chile ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeChileIdCardRequest()
        request.body = ChileIdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_chile_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Chile ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeChileIdCardRequest()
        request.body = ChileIdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_chile_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Chile ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeChileIdCardRequest{}
    imageChileIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.ChileIdCardRequestBody{
        Image: &imageChileIdCardRequestBody,
    }
    response, err := client.RecognizeChileIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Chile ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeChileIdCardRequest{}
    urlChileIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.ChileIdCardRequestBody{
        Url: &urlChileIdCardRequestBody,
    }
    response, err := client.RecognizeChileIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.11 Vietnam ID Card

Function

This API detects and extracts text from images of Vietnam-issued ID cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only ID cards issued by Vietnam can be recognized.
- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card can be rotated to any angle.
- Illuminated or dark images, or images with anti-counterfeit watermarks can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/vietnam-id-card

Table 4-77 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-78 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

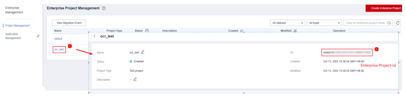
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-79 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data. ● The URL cannot contain Chinese characters. If Chinese characters exist, they must be encoded using UTF-8.

Parameter	Mandatory	Type	Description
side	No	String	Whether the front or back side of the ID card is displayed. The options are as follows: ● **front**: front of the Vietnam ID card ● **back**: back of the Vietnam ID card If the value of this parameter is empty or not included, the system will automatically recognize whether the image is the front or back of an ID card. It is recommended to set this parameter for higher accuracy.
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: ● **true**: The Base64 code of the portrait on the ID card will be returned. ● **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the ID card. The options are as follows: ● **true**: The location of the portrait on the ID card will be returned. ● **false**: The location of the portrait on the ID card will not be returned.

Parameter	Mandatory	Type	Description
return_idcard_type	No	Boolean	Whether to return the ID card type. The options are as follows: ● **true**: The ID card type will be returned, indicating that the ID card is the original ID card or copy of the original ID card. ● **false**: The ID card type will not be returned.
return_text_location	No	Boolean	Location of a text block. The options are as follows: ● **true**: All text blocks will be returned. ● **false**: Text blocks will not be returned. If this parameter is not specified, the system does not return the location of any text blocks by default. If a non-Boolean value is entered, an error message will be displayed, indicating that the parameter is invalid.

Response Parameters

Status code: 200

Table 4-80 Response body parameter

Parameter	Type	Description
result	**Table 4-81** object	Calling result This parameter is not returned when the API fails to be called.

Table 4-81 VietnamIdCardResult

Parameter	Type	Description
side	String	Side of the ID card. The value can be **front** or **back**.
number	String	Card number

Parameter	Type	Description
full_name	String	Name
birth_date	String	Date of birth
sex	String	Gender
nationality	String	Nationality
origin_place	String	Place of origin
residence_place	String	Residence
expiry_date	String	Validity period
personal_identification	String	Personal identification. This field is returned when **side** is set to **back**.
issue_date	String	Date of issue. This field is returned when **side** is set to **back**.
machine_code1	String	Machine-readable code in the first line on the back of an ID card. This field is returned when **side** is set to **back**.
machine_code2	String	Machine-readable code in the second line on the back of an ID card. This field is returned when **side** is set to **back**.
machine_code3	String	Machine-readable code in the third line on the back of an ID card. This field is returned when **side** is set to **back**.
confidence	Object	Confidence of related fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
portrait_image	String	Base64 code of the portrait when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image when **return_portrait_location** is set to **true**, including the two-dimensional coordinates (x,y) of the four vertices of the portrait, which is displayed in a list. The origin is at the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
idcard_type	String	ID card type returned when **return_idcard_type** is set to **true**. **normal** indicates the original ID card, **copy** indicates the copy of the ID card, and **screen** indicates a recapture of an ID card image on the screen.

Parameter	Type	Description
text_location	Object	Location of all fields identified on the original image, including the two-dimensional coordinates (x,y) of the four vertices in all text areas. The image coordinate system is used. The coordinate origin is the upper left corner of the image, the X axis is horizontal, and the Y axis is vertical.

Status code: 400

Table 4-82 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- The endpoint is the request URL for calling an API. Endpoints vary according to services and regions. For details, see **Endpoints**. For example, if Vietnam ID Card OCR is deployed in the **CN-Hong Kong** region, the endpoint is **ocr.ap-southeast-1.myhuaweicloud.com**, and the request URL is **https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/vietnam-id-card**. **project_id** indicates the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

```
POST https://{endpoint}/v2/{project_id}/ocr/vietnam-id-card

{
  "image" : "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "side" : "front",
    "number" : "784-1995-xxxxxxx-4",
    "full_name" : "VŨ THỊ HẢI YẾN",
    "birth_date" : "08-08-1980",
    "sex" : "Nữ",
    "nationality" : "Việt Nam",
    "origin_place" : "Giới Phiên Thành phố Yên Bái Yên Bải",
    "residence_place" : "Thôn Ngòi Châu Giới Phiên Thành phố Yên Bái Yên Bái",
    "expiry_date" : "08/07/2030",
    "confidence" : {
      "number" : 0.9993,
      "full_name" : 0.9983,
      "birth_date" : 0.9998,
      "sex" : 0.9999,
      "nationality" : 0.9999,
      "origin_place" : 0.9855,
      "residence_place" : 0.9984,
      "expiry_date" : 0.9995
    },
    "idcard_type" : "normal",
    "portrait_image" : "/9j/4AAQSkZJRgABAQAAAQABAA...",
    "portrait_location" : [ [ 217, 359 ], [ 582, 360 ], [ 573, 882 ], [ 199, 885 ] ]
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "AIS.0103",
  "error_msg" : "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Vietnam ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeVietnamIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeVietnamIdCardRequest request = new RecognizeVietnamIdCardRequest();
        VietnamIdCardRequestBody body = new VietnamIdCardRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeVietnamIdCardResponse response = client.recognizeVietnamIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Vietnam ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeVietnamIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeVietnamIdCardRequest request = new RecognizeVietnamIdCardRequest();
        VietnamIdCardRequestBody body = new VietnamIdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeVietnamIdCardResponse response = client.recognizeVietnamIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Vietnam ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeVietnamIdCardRequest()
        request.body = VietnamIdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_vietnam_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Vietnam ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
```

```
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeVietnamIdCardRequest()
        request.body = VietnamIdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_vietnam_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Vietnam ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeVietnamIdCardRequest{}
    imageVietnamIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.VietnamIdCardRequestBody{
        Image: &imageVietnamIdCardRequestBody,
    }
    response, err := client.RecognizeVietnamIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Vietnam ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeVietnamIdCardRequest{}
    urlVietnamIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.VietnamIdCardRequestBody{
        Url: &urlVietnamIdCardRequestBody,
    }
    response, err := client.RecognizeVietnamIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.12 Peru ID Card

Function

This API detects and extracts text from images of Peru-issued identity cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-6 Peru ID card example 1

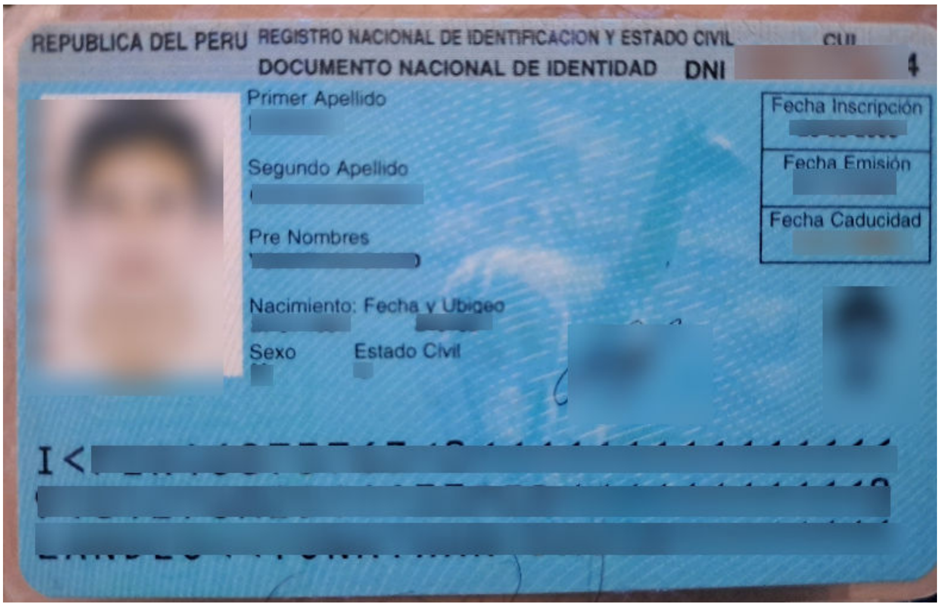
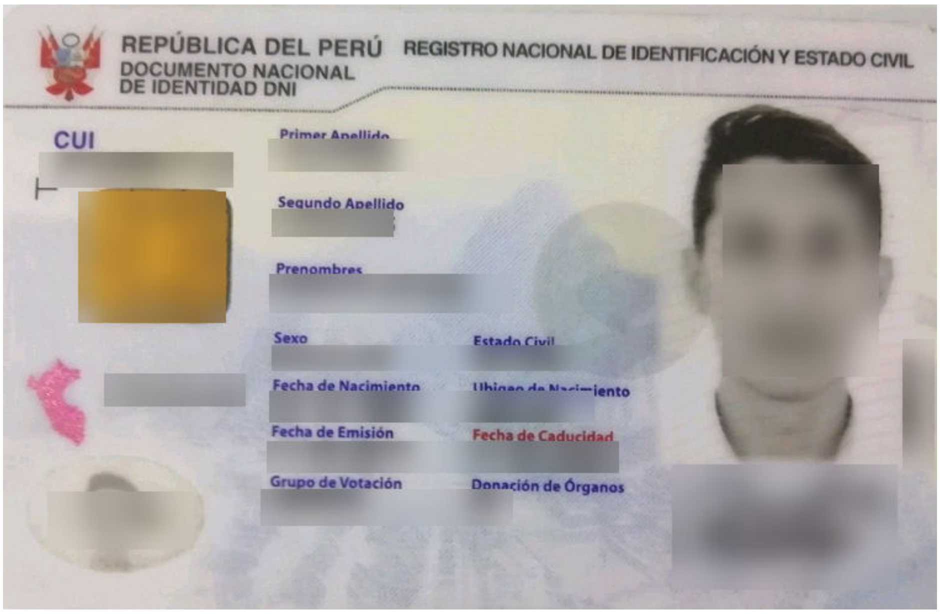


Figure 4-7 Peru ID card example 2



Constraints and Limitations

- Only ID cards issued by Peru can be recognized.
- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card to be recognized must occupy more than 25% of the image. When scanning an ID card, ensure that the entire ID card is displayed in the image.
- An ID card can be rotated to any angle.
- The ID card in the image can be moderately distorted, but the aspect ratio cannot be distorted by more than 10%.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/peru-id-card

Table 4-83 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-84 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

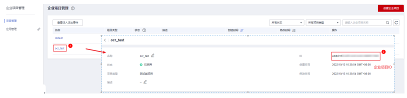
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR allows you to use Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  企业项目管理 项目管理 应用管理 企业项目管理 创建企业项目 查看迁入迁出事件 所有状态 所有项目类型 请输入企业项目名称 名称 项目类型 状态 描述 创建时间 修改时间 操作 default ocr_test 1 2 企业项目ID ocr_test 名称 ocr_test 状态 已启用 项目类型 测试类项目 描述 -- ID eddb01 创建时间 2022/10/13 10:38:54 GMT+08:00 修改时间 2022/10/13 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-85 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. PDFs with multiple pages can also be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: ● **true**: The Base64 encoded string of the portrait on the ID card will be returned. ● **false**: A null value is returned. If this parameter is not set, the default value **false** is used, and a null value is returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the ID card. The options are as follows: ● **true**: The location of the portrait on the ID card will be returned. ● **false**: A null value is returned. If this parameter is not set, the default value **false** is used, and a null value is returned.

Response Parameters

Status code: 200

Table 4-86 Response body parameter

Parameter	Type	Description
result	**PeruIdCardResult** object	Result of a successful API call. This parameter is not included for a failed call.

Table 4-87 PeruIdCardResult

Parameter	Type	Description
cui_number	String	ID number
first_surname	String	First surname
second_surname	String	Second surname
given_name	String	Other names

Parameter	Type	Description
sex	String	Gender
marital_status	String	Marital status
birth_date	String	Date of birth
nationality	String	Nationality
issue_date	String	Date of issue
expiry_date	String	Date of expiration
birth_place	String	Birthplace code
voting_group	String	Voting group
organ_donation	String	Organ donation
registration_date	String	Registration date
portrait_image	String	Base64 encoded string of the portrait. This parameter is returned only when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
address	String	Address
department	String	Department
province	String	Province
district	String	District
remarks	String	Remarks
machine_code1	String	Machine code in the first line
machine_code2	String	Machine code in the second line
machine_code3	String	Machine code in the third line

Parameter	Type	Description
duplicate	Boolean	Whether the ID card is a duplicate. The options are as follows: ● **true**: The ID card is a duplicate. ● **false**: The ID card is not a duplicate.
confidence	Map<String,Number>	Confidence of related fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Status code: 400

Table 4-88 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned for a successful call.
error_msg	String	Error message when the API call fails This parameter is not returned for a successful call.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Peru ID Card OCR is deployed in the **LA-Santiago** region. The endpoint is **ocr.la-south-2.myhuaweicloud.com** or **ocr.la-south-2.myhuaweicloud.cn**. The request URL is **https://ocr.la-south-2.myhuaweicloud.com/v2/{project_id}/peru-id-card**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the Base64 encoded string of an image.)

```
POST https://{endpoint}/v2/{project_id}/ocr/peru-id-card

{
  "image" : "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "return_portrait_image" : true,
  "return_portrait_location" : true
}
```

- Example request (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/peru-id-card

{
  "url" : "https://BucketName.obs.myhuaweicloud.com/ObjectName",
  "return_portrait_image" : true,
  "return_portrait_location" : true
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "cui_number" : "4687xxx3-4",
    "first_surname" : "LANDEO",
    "second_surname" : "CARHUALLANQUI",
    "given_name" : "YONATHAN DAVID",
    "sex" : "M",
    "marital_status" : "C",
    "birth_date" : "xx 01 1991",
    "nationality" : "xx",
    "issue_date" : "31 12 xx21",
    "expiry_date" : "13 11 2025",
    "birth_place" : 240102,
    "voting_group" : "xx",
    "organ_donation" : "xx",
    "registration_date" : "26 03 2009",
    "portrait_image" : "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDA...",
    "portrait_location" : [ [ 25, 55 ], [ 138, 55 ], [ 139, 216 ], [ 26, 217 ] ],
    "address" : "xxx",
    "department" : "xxx",
    "province" : "xxx",
    "district" : "xxx",
    "remarks" : "xxx",
    "machine_code1" : "I<PER46873763<0<<<<<<<<<<<<<<<",
    "machine_code2" : "9101276M2511133PER<<<<<<<<<<<0",
    "machine_code3" : "LANDEO<<YONATHAN<DAVID<<<<<<<<",
    "duplicate" : false,
    "confidence" : {
      "cui_number" : 0.9282,
      "first_surname" : 0.9016,
      "second_surname" : 0.9309,
      "given_name" : 0.9307,
      "sex" : 0.8633,
      "marital_status" : 0.6044,
      "birth_date" : 0.8828,
      "nationality" : 0.8826,
      "issue_date" : 0.9226,
      "expiry_date" : 0.9158,
      "birth_place" : 0.9122,
      "voting_group" : 0.8826,
      "organ_donation" : 0.8826,
      "registration_date" : 0.913,
      "portrait_image" : 0.945,
      "portrait_location" : 0.994,
      "address" : 0.9021,
      "department" : 0.8973,
      "province" : 0.8769,
      "district" : 0.8647,
      "remarks" : 0.8476,
      "machine_code1" : 0.8384,
      "machine_code2" : 0.9363,
      "machine_code3" : 0.9046,
      "duplicate" : 0.8703
    }
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "AIS.0103",
  "error_msg" : "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 encoded string of the Peru ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizePeruIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizePeruIdCardRequest request = new RecognizePeruIdCardRequest();
        PeruIdCardRequestBody body = new PeruIdCardRequestBody();
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizePeruIdCardResponse response = client.recognizePeruIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Peru ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizePeruIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizePeruIdCardRequest request = new RecognizePeruIdCardRequest();
        PeruIdCardRequestBody body = new PeruIdCardRequestBody();
        body.withReturnPortraitLocation(true);
        body.withReturnPortraitImage(true);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizePeruIdCardResponse response = client.recognizePeruIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 encoded string of the Peru ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizePeruIdCardRequest()
        request.body = PeruIdCardRequestBody(
            return_portrait_location=True,
            return_portrait_image=True,
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_peru_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Peru ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizePeruIdCardRequest()
        request.body = PeruIdCardRequestBody(
            return_portrait_location=True,
            return_portrait_image=True,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_peru_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 encoded string of the Peru ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizePeruIdCardRequest{}
    returnPortraitLocationPeruIdCardRequestBody:= true
    returnPortraitImagePeruIdCardRequestBody:= true
    imagePeruIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.PeruIdCardRequestBody{
        ReturnPortraitLocation: &returnPortraitLocationPeruIdCardRequestBody,
        ReturnPortraitImage: &returnPortraitImagePeruIdCardRequestBody,
        Image: &imagePeruIdCardRequestBody,
    }
    response, err := client.RecognizePeruIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Peru ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizePeruIdCardRequest{}
    returnPortraitLocationPeruIdCardRequestBody:= true
    returnPortraitImagePeruIdCardRequestBody:= true
    urlPeruIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.PeruIdCardRequestBody{
        ReturnPortraitLocation: &returnPortraitLocationPeruIdCardRequestBody,
        ReturnPortraitImage: &returnPortraitImagePeruIdCardRequestBody,
        Url: &urlPeruIdCardRequestBody,
    }
    response, err := client.RecognizePeruIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Successful response
400	Failed response

See **Status Codes**.

Error Codes

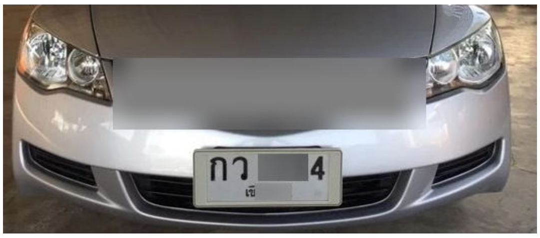
See **Error Codes**.

4.13 Thailand Plate Number

Function

This API detects and extracts license plate information from images of Thailand license plates and returns the license plate number and location. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Figure 4-8 Example Thailand plate number



Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- A license plate can be rotated to any angle.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/thailand-license-plate

Table 4-89 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-90 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

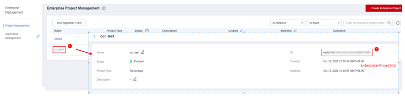
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: - If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. - If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. - If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-91 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64 encoded string of an image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, TIFF, or JPEG format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Response Parameters

Status code: 200

Table 4-92 Response body parameter

Parameter	Type	Description
result	Array of **ThailandLicensePlateItem** objects	Calling result This parameter is not returned when the API fails to be called.

Table 4-93 ThailandLicensePlateItem

Parameter	Type	Description
plate_number	String	License plate content
plate_location	Array<Array<Integer>>	List of location information about a license plate, including the 2D coordinates (x, y) of four vertexes in the text area, where the coordinate origin is the upper-left corner of the image, theX axis is horizontal, and the Y axis is vertical.
confidence	Float	Confidence of a field. The value ranges from 0 to 1. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.
province	String	Province where the license plate is registered

Status code: 400

Table 4-94 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

NOTE

- **endpoint** is the request URL for calling an API. Endpoints vary depending on services and regions. For details, see **Endpoints**.

  For example, Thailand Plate Number OCR is deployed in the **AP-Bangkok** region. The endpoint is **ocr.ap-southeast-2.myhuaweicloud.com** or **ocr.ap-southeast-2.myhuaweicloud.cn**. The request URL is **https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/thailand-license-plate**. **project_id** is the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
- For details about how to obtain a token, see **Making an API Request**.

- Request example (Method 1: Use the image Base64 string.)

```
POST https://{endpoint}/v2/{project_id}/ocr/thailand-license-plate

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
  "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
}
```

- Request example (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/thailand-license-plate

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
  "url":"https://BucketName.obs.xxxx.com/ObjectName"
}
```

- Sample code for a Python 3 request (For codes in other languages, refer to the following sample or use OCR SDK.)

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v2/{project_id}/ocr/thailand-license-plate"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/thailand-license-plate-demo.png' # Read a local image.
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use the Base64 code of the image.
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : [ {
    "plate_number" : "กง XXX4",
    "province" : "มหาสารคาม",
    "confidence" : 0.9225,
    "plate_location" : [ [ 370, 881 ], [ 2591, 881 ], [ 2591, 2281 ], [ 370, 2281 ] ]
  } ]
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 code of the Thailand plate number image image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeThailandLicensePlateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeThailandLicensePlateRequest request = new RecognizeThailandLicensePlateRequest();
        ThailandLicensePlateRequestBody body = new ThailandLicensePlateRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeThailandLicensePlateResponse response = 
client.recognizeThailandLicensePlate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Thailand plate number image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeThailandLicensePlateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeThailandLicensePlateRequest request = new RecognizeThailandLicensePlateRequest();
        ThailandLicensePlateRequestBody body = new ThailandLicensePlateRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeThailandLicensePlateResponse response =
client.recognizeThailandLicensePlate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 code of the Thailand plate number image image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeThailandLicensePlateRequest()
        request.body = ThailandLicensePlateRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_thailand_license_plate(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Thailand plate number image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeThailandLicensePlateRequest()
        request.body = ThailandLicensePlateRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_thailand_license_plate(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 code of the Thailand plate number image image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeThailandLicensePlateRequest{}
    imageThailandLicensePlateRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.ThailandLicensePlateRequestBody{
        Image: &imageThailandLicensePlateRequestBody,
    }
    response, err := client.RecognizeThailandLicensePlate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Thailand plate number image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeThailandLicensePlateRequest{}
    urlThailandLicensePlateRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.ThailandLicensePlateRequestBody{
        Url: &urlThailandLicensePlateRequestBody,
    }
    response, err := client.RecognizeThailandLicensePlate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.14 Hong Kong ID Card

Function

This API detects and extracts text from images of Hong Kong (China)-issued ID cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card to be recognized must occupy more than 50% of the image. When scanning an ID card, ensure that the entire ID card is displayed in the image.
- Images can be rotated horizontally to any angle, but the recognition precision is affected.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/hk-id-card

Table 4-95 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.

Parameter	Mandatory	Description
project_id	Yes	Project ID, which can be obtained from **Obtaining a Project ID**.

Request Parameters

Table 4-96 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

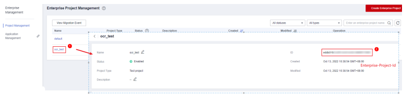
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: - If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. - If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. - If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-97 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: - **true**: The Base64 code of the portrait on the ID card will be returned. - **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the ID card. The options are as follows: - **true**: The location of the portrait on the ID card will be returned. - **false**: The location of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the location of the portrait on the ID card will not be returned.

Response Parameters

Status code: 200

Table 4-98 Response body parameter

Parameter	Type	Description
result	**HkIdCardResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-99 HkIdCardResult

Parameter	Type	Description
name	String	Chinese name
name_en	String	Name in English
sex	String	Gender **M**: male; **F**: female
birth_date	String	Date of birth
number	String	ID number
symbols	String	Symbol on the ID card
name_telegraph_code	String	Name in Chinese telegraph code
permanent	Boolean	Whether the ID card is a Hong Kong permanent ID card. The options are as follows: **true**: permanent; **false**: non-permanent
initial_issue_date	String	Date of first issue
issue_date	String	Date of issue
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
portrait_image	String	Base64-encoded profile image. The profile with a larger size is returned by default. This parameter is available only when **return_portrait_image** is set to **true**.
confidence	Object	Confidence of each field

Status code: 400

Table 4-100 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

- Request example (Method 1: Use the image Base64 string.)

```
POST https://ocr.ap-southeast-1.myhuaweicloud.com/v2/<project_id>/ocr/hk-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
 }
```

- Request example (Method 2: Use the image URL.)

```
POST https://ocr.ap-southeast-1.myhuaweicloud.com/v2/<project_id>/ocr/hk-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "url":"https://BucketName.obs.myhwclouds.com/ObjectName"
 }
```

Example Response

Status code: 200

Example response for a successful request

```
{
   "result": {
      "name": "Zhang San",
      "name_en": "ZHANG, SAN",
      "sex": "F",
      "birth_date": "01-01-1988",
      "number": "C66xxxx(E)",
      "symbols": "***AZ",
      "name_telegraph_code": "39xxxxxxx174",
      "permanent": true,
      "initial_issue_date": "(01-99)",
      "issue_date": "15-09-18",
      "confidence": {
         "name": 0.8876,
         "name_en": 0.9988,
         "sex": 0.9997,
         "birth_date": 0.9993,
         "number": 0.9832,
         "symbols": 0.959,
         "name_telegraph_code": 0.9998,
         "permanent": 0.9223,
         "initial_issue_date": 0.963,
         "issue_date": 0.9998
      },
      "portrait_location": [
         [
             35,
             115
         ],
         [
             154,
             115
         ],
         [
             153,
             263
         ],
         [
             34,
             263
         ]
      ]
   }
}
```

Status code: 400

Example response for a failed request

```
{
    "error_code": "AIS.0103",
    "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 encoded string of the Hong Kong (China) ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeHkIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeHkIdCardRequest request = new RecognizeHkIdCardRequest();
        HkIdCardRequestBody body = new HkIdCardRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeHkIdCardResponse response = client.recognizeHkIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Hong Kong (China) ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeHkIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeHkIdCardRequest request = new RecognizeHkIdCardRequest();
        HkIdCardRequestBody body = new HkIdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeHkIdCardResponse response = client.recognizeHkIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 encoded string of the Hong Kong (China) ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeHkIdCardRequest()
        request.body = HkIdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_hk_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Hong Kong (China) ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

```
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeHkIdCardRequest()
        request.body = HkIdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_hk_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 encoded string of the Hong Kong (China) ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeHkIdCardRequest{}
    imageHkIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.HkIdCardRequestBody{
        Image: &imageHkIdCardRequestBody,
    }
    response, err := client.RecognizeHkIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Hong Kong (China) ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeHkIdCardRequest{}
    urlHkIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.HkIdCardRequestBody{
        Url: &urlHkIdCardRequestBody,
    }
    response, err := client.RecognizeHkIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request

Status Code	Description
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.15 Macao Resident ID Card

Function

This API detects and extracts text from images of Macao (China)-issued resident ID cards and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, JPEG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- An ID card to be recognized must occupy more than 50% of the image. When scanning an ID card, ensure that the entire ID card is displayed in the image.
- Images can be rotated horizontally to any angle, but the recognition precision is affected.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/macao-id-card

Table 4-101 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-102 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

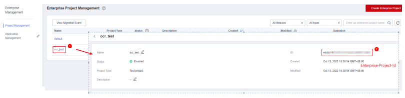
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: ● If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. ● If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. ● If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-103 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPEG, JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
side	No	String	**front**: front of an ID card **back**: back of an ID card If this parameter is left empty or not included, the system automatically recognizes whether the image is the front or back of an ID card. To ensure accuracy, set this parameter.
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: ● **true**: The Base64 code of the portrait on the ID card will be returned. ● **false**: The Base64 code of the portrait on the ID card will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the ID card will not be returned.

Response Parameters

Status code: 200

Table 4-104 Response body parameter

Parameter	Type	Description
result	**MacaoIdCardResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-105 MacaoIdCardResult

Parameter	Type	Description
side	String	Whether the front or back side of the ID card is displayed. The options are as follows: ● **front**: front side ● **back**: back side

Parameter	Type	Description
name	String	Name
name_en	String	Name in English. Words in the name are separated by spaces.
sex	String	Gender. The options are **M** and **F**. **M** indicates male, and **F** indicates female.
issue_date	String	Date of issue
expiry_date	String	Date of expiry
birth_date	String	Date of birth
initial_issue_date	String	Date of first issue
height	String	Height
number	String	ID number
symbols	String	Code for place of birth and gender
machine_code1	String	Machine-readable code in the first line on the back of an ID card
machine_code2	String	Machine-readable code in the second line on the back of an ID card
machine_code3	String	Machine-readable code in the third line on the back of an ID card
portrait_image	String	Base64 code of the head portrait on the ID card. If the input parameter **return_portrait_image** is missing, this parameter does not exist.
confidence	Object	Confidence of related fields. A higher confidence indicates a higher accuracy of the field identified. The confidence is calculated using algorithms and is not equal to the accuracy.

Status code: 400

Table 4-106 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. This parameter is not included when the API is successfully called.

Parameter	Type	Description
error_msg	String	Error message when the API call fails. This parameter is not included when the API is successfully called.

Example Request

- Request example (Method 1: Use the image Base64 string.)

```
POST https://{endpoint}/v2/{project_id}/ocr/macao-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
    "side": "front",
    "return_portrait_image": "false"
 }
```

- Request example (Method 2: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/ocr/macao-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.myhwclouds.com/ObjectName",
    "side": "front",
    "return_portrait_image": "false"
 }
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "id_number" : "784-1995-xxxxxxx-4",
    "name": "Zhang San",
    "name_en" : "ZHANG, SAN",
    "birth_date" : "08-08-1980",
    "sex" : "M",
    "issue_date" : "08-08-2012",
    "expiry_date" : "08-08-2022",
    "initial_issue_date" : "18-02-2000",
    "height" : "166",
    "symbols" : "AS",
    "confidence" : {
      "id_number" : 0.9234,
      "name" : 0.9234,
      "name_en" : 0.9234,
      "birth_date" : 0.9234,
      "sex" : 0.9234,
      "issue_date" : 0.9234,
      "expiry_date" : 0.9234,
      "initial_issue_date" : 0.9234,
      "height" : 0.9234,
      "symbols" : 0.9234
    }
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "AIS.0103",
  "error_msg" : "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 encoded string of the Macao (China) ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMacaoIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMacaoIdCardRequest request = new RecognizeMacaoIdCardRequest();
        MacaoIdCardRequestBody body = new MacaoIdCardRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeMacaoIdCardResponse response = client.recognizeMacaoIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Macao (China) ID card image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMacaoIdCardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMacaoIdCardRequest request = new RecognizeMacaoIdCardRequest();
        MacaoIdCardRequestBody body = new MacaoIdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeMacaoIdCardResponse response = client.recognizeMacaoIdCard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 encoded string of the Macao (China) ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMacaoIdCardRequest()
        request.body = MacaoIdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_macao_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Macao (China) ID card image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMacaoIdCardRequest()
        request.body = MacaoIdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_macao_id_card(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 encoded string of the Macao (China) ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMacaoIdCardRequest{}
    imageMacaoIdCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.MacaoIdCardRequestBody{
        Image: &imageMacaoIdCardRequestBody,
    }
    response, err := client.RecognizeMacaoIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Macao (China) ID card image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMacaoIdCardRequest{}
    urlMacaoIdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.MacaoIdCardRequestBody{
        Url: &urlMacaoIdCardRequestBody,
    }
    response, err := client.RecognizeMacaoIdCard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.16 Exit-Entry Travel Permit

Function

This API detects and extracts text from images of Exit-Entry Permit for Traveling to and from Hong Kong, Macao, and Taiwan and converts the text into a structured format. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- A travel permit to be recognized must occupy more than 50% of the image. When scanning a travel permit, ensure that the entire travel permit is displayed in the image.
- Images can be rotated horizontally to any angle, but the recognition precision is affected.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/exit-entry-permit

Table 4-107 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-108 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

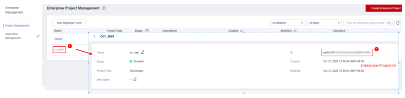
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: - If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. - If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. - If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-109 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image file. The image file has a size limit of 10 MB. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg....** If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: • **true**: The Base64 code of the portrait on the travel permit will be returned. • **false**: The Base64 code of the portrait on the travel permit will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the travel permit will not be returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the travel permit. The options are as follows: • **true**: The location of the portrait on the travel permit will be returned. • **false**: The location of the portrait on the travel permit will not be returned. If this parameter is not specified, **false** is used by default.

Response Parameters

Status code: 200

Table 4-110 Response body parameter

Parameter	Type	Description
result	**ExitEntryPermitResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-111 ExitEntryPermitResult

Parameter	Type	Description
name	String	Name
name_en	String	Name in English
sex	String	Gender
birth_date	String	Date of birth
number	String	Permit number
issuing_authority	String	Issuing authority
issue_place	String	Place where the permit is issued
valid_period	String	Date of expiry
machine_code	String	Machine-readable code
portrait_image	String	Base64 code of the portrait. This parameter is returned only when **return_portrait_image** is set to **true**.
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
type	String	Permit type. The options are as follows: ● **Exit-Entry Permit for Traveling to and from Hong Kong and Macao** ● **Exit-Entry Permit for Traveling to and from Taiwan**
side	String	Whether the front or back of the travel permit is displayed. The options are as follows: ● **front**: front side ● **back**: back side
endorsement_info_hk	**ExitEntryPermitEndorsementInfo** object	Hong Kong (China) endorsement information. This field is returned only when the permit type is **Exit-Entry Permit for Traveling to and from Hong Kong and Macao**.

Parameter	Type	Description
endorsement_info_mo	**ExitEntryPermitEndorsementInfo** object	Macao (China) endorsement information. This field is returned only when the permit type is **Exit-Entry Permit for Traveling to and from Hong Kong and Macao**.
endorsement_info_tw	**ExitEntryPermitEndorsementInfo** object	Taiwan (China) endorsement information. This field is returned only when the permit type is **Exit-Entry Permit for Traveling to and from Taiwan**.
confidence	**ExitEntryPermitConfidence** object	Confidence of each field

Table 4-112 ExitEntryPermitEndorsementInfo

Parameter	Type	Description
endorsement_type	String	Endorsement type
valid_round_trips	String	Number of valid round journeys on the endorsement
endorsement_valid_period	String	Date of expiry of the endorsement
remark	String	Endorsement remarks
issue_info	String	Endorsement issuing information

Table 4-113 ExitEntryPermitConfidence

Parameter	Type	Description
name	Float	Confidence of the name
name_en	Float	Confidence of the English name
birth_date	Float	Confidence of the birth date
sex	Float	Confidence of the gender
number	Float	Confidence of the permit number
valid_period	Float	Confidence of the validity period
issuing_authority	Float	Confidence of the issuing authority
issue_place	Float	Confidence of the place of issuance

Parameter	Type	Description
machine_code	Float	Confidence of the machine code
type	Float	Confidence of a permit type
side	Float	Confidence of information on the front and back sides of image of the permit
endorsement_info_hk	Object	Confidence of the Hong Kong (China) endorsement information
endorsement_info_mo	Object	Confidence of the Macao (China) endorsement information
endorsement_info_tw	Object	Confidence of Taiwan (China) endorsement information

Status code: 400

Table 4-114 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. If error code **ModelArts.4204** is displayed, refer to **Why Is a Message Stating "ModelArts.4204" Displayed When the OCR API Is Called?** This parameter is not returned when the API is successfully called.
error_msg	String	Error message when the API call fails This parameter is not returned when the API is successfully called.

Example Request

- Request example (Method 1: Use the image Base64 string.)

```
POST https://ocr.cn-north-1.myhuaweicloud.com/v2/<project_id>/ocr/exit-entry-permit
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
   "image":"/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
 }
```

- Request example (Method 2: Use the image URL.)

```
POST https://ocr.cn-north-1.myhuaweicloud.com/v2/<project_id>/ocr/exit-entry-permit
Request Header:
Content-Type: application/json
```

```
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url":"https://BucketName.obs.myhwclouds.com/ObjectName"
 }
```

Example Response

Status code: 200

Example response for a successful request (the front)

```
{
    "result":{
        "name": "Zhang San",
        "name_en": "ZHANG, SAN",
        "birth_date": "1970.12.21",
        "sex":"M",
        "number": "C319XXXXX",
        "valid_period": "2016.01.04 - 2026.01.03",
        "issuing_authority": "Bureau of Exit and Entry Administration of the Ministry of Public Security",
        "issue_place": "Guangdong",
        "machine_code": "CSCA3xxxxxxx<290xxxx<810xxxx<2",
        "type": "Exit-Entry Permit for Traveling to and from Hong Kong and Macao",
        "side": "front",
        "portrait_image": "/9j/4AAQSkZJRg…",
        "portrait_location": [
            [
                1174,
                1326
            ],
            [
                1378,
                1294
            ],
            [
                1416,
                1534
            ],
            [
                1212,
                1566
            ]
      ],
        "confidence": {
            "name": 0.9254,
            "name_en": 0.9245,
            "birth_date": 0.9014,
            "sex": 0.9896,
            "number": 0.9741,
            "valid_period": 0.9542,
            "issuing_authority": 0.9635,
            "issue_place": 0.9749,
            "machine_code": 0.8945,
            "type": 0.9650,
            "side": 0.9785
        }
    }
}
```

Example response for a successful request (the back)

```
{
    "result":{
        "type": "Exit-Entry Permit for Traveling to and from Hong Kong and Macao",
        "side": "back",
        "endorsement_info_hk": {
            "endorsement_type": "Individual visits (G)",
            "valid_round_trips": "Valid for double journey per visit",
            "endorsement_valid_period": "2014.04.21-2015.04.20",
            "remark": "Stay for no more than 7 days in Hong Kong (China)",
            "issue_info": "4400-H001"
        },
        "endorsement_info_mo": {
            "endorsement_type": "Individual visits (G)",
            "valid_round_trips": "Valid for double journey per visit",
            "endorsement_valid_period": "2014.04.21-2015.04.20",
            "remark": "Stay for no more than 7 days in Macao (China)",
            "issue_info": "4400-M001"
        },
        "confidence": {
            "type": 0.9650,
            "side": 0.9785,
            "endorsement_info_hk": {
            "endorsement_type": 0.9214,
            "valid_round_trips": 0.9852,
            "endorsement_valid_period": 0.9474,
            "remark": 0.9996,
            "issue_info": 0.9524
        },
        "endorsement_info_mo": {
            "endorsement_type": 0.9741,
            "valid_round_trips": 0.9598,
            "endorsement_valid_period": 0.9157,
            "remark": 0.9145,
            "issue_info": 0.9935
        }
    }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code": "AIS.0103",
  "error_msg": "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 encoded string of the exit-entry travel permit image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeExitEntryPermitSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeExitEntryPermitRequest request = new RecognizeExitEntryPermitRequest();
        ExitEntryPermitRequestBody body = new ExitEntryPermitRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeExitEntryPermitResponse response = client.recognizeExitEntryPermit(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the exit-entry travel permit image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeExitEntryPermitSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeExitEntryPermitRequest request = new RecognizeExitEntryPermitRequest();
        ExitEntryPermitRequestBody body = new ExitEntryPermitRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeExitEntryPermitResponse response = client.recognizeExitEntryPermit(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 encoded string of the exit-entry travel permit image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeExitEntryPermitRequest()
        request.body = ExitEntryPermitRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_exit_entry_permit(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the exit-entry travel permit image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeExitEntryPermitRequest()
        request.body = ExitEntryPermitRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_exit_entry_permit(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 encoded string of the exit-entry travel permit image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeExitEntryPermitRequest{}
    imageExitEntryPermitRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.ExitEntryPermitRequestBody{
        Image: &imageExitEntryPermitRequestBody,
    }
    response, err := client.RecognizeExitEntryPermit(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the exit-entry travel permit image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeExitEntryPermitRequest{}
    urlExitEntryPermitRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.ExitEntryPermitRequestBody{
        Url: &urlExitEntryPermitRequestBody,
    }
    response, err := client.RecognizeExitEntryPermit(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

4.17 Chinese Mainland Travel Permit

Function

This API detects and extracts text from images of Chinese Mainland Travel Permit for Hong Kong (China), Macao (China), and Taiwan (China) Residents and converts the text into a structured format. This OCR service supports the following two types of Chinese mainland travel permits: Chinese mainland travel permit for Hong Kong (China) and Macao (China) residents and Chinese mainland travel permit for Taiwan (China) residents. For details about the constraints on using this API, see **Constraints and Limitations**. For details about how to use this API, see **Introduction to OCR**.

Constraints and Limitations

- Only images in PNG, JPG, BMP, or TIFF format can be recognized.
- No side of the image can be smaller than 15 or larger than 8,192 pixels.
- A travel permit to be recognized must occupy more than 50% of the image. When scanning a travel permit, ensure that the entire travel permit is displayed in the image.
- Images can be rotated horizontally to any angle, but the recognition precision is affected.
- Illuminated or dark images can be recognized, but the accuracy may be compromised.

Calling Method

For details, see **Calling APIs**.

Prerequisites

Before using this API, subscribe to the service and complete authentication. For details, see **Subscribing to an OCR Service** and **Authentication**.

NOTE

Before you use the service for the first time, subscribe to the service by clicking **Subscribe**. You only need to subscribe to the service once. If you have not subscribed to the service yet, error "ModelArts.4204" will be displayed when you call this API. Before you call the API, log in to the OCR console and subscribe to the corresponding service. Ensure that you make the subscription to the service in the same region where you want to call this API.

URI

POST /v2/{project_id}/ocr/mainland-travel-permit

Table 4-115 URI parameters

Parameter	Mandatory	Description
endpoint	Yes	Endpoint, which is the request address for calling an API. The endpoint varies depending on services in different regions. For more details, see **Endpoints**.
project_id	Yes	Project ID, which can be obtained by referring to **Obtaining a Project ID**.

Request Parameters

Table 4-116 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. The token is the value of **X-Subject-Token** in the response header in **Authentication**.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

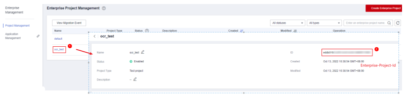
Parameter	Mandatory	Type	Description
Enterprise-Project-Id	No	String	Enterprise project ID. OCR uses Enterprise Project Management Service (EPS) to split fees for resources used by different user groups and users. To obtain the enterprise project ID, go to the **Enterprise Project Management** console, click the enterprise project name, and obtain the enterprise project ID on the enterprise project details page.  Enterprise Management Project Management Application Management Enterprise Project Management Create Enterprise Project View Migration Event All statuses All types Enter an enterprise project name. Name Project Type Status Description Created Modified Operation default ocr_test 1 2 Enterprise-Project-Id ocr_test Name ocr_test Status Enabled Project Type Test project Description -- ID eddb016[redacted] Created Oct 13, 2022 10:38:54 GMT+08:00 Modified Oct 13, 2022 10:38:54 GMT+08:00 For details about how to create an enterprise project, see *Optical Character Recognition User Guide*. **NOTE** After an enterprise project is created, parameter transfer involves the following scenarios: - If a correct enterprise project ID is carried and the OCR service can be used properly, the bills are allocated to the enterprise project corresponding to the ID. - If an incorrect enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**. - If no enterprise project ID is carried and the OCR service can be used properly, the enterprise project in the bills is allocated to **Non-project**.

Table 4-117 Request body parameters

Parameter	Mandatory	Type	Description
image	No	String	Set either this parameter or **url**. Base64-encoded image data. No side of the image can be smaller than 15 or larger than 8,192 pixels. Only images in JPG, PNG, BMP, or TIFF format can be recognized. An example is **/9j/4AAQSkZJRgABAg...**. If the image data contains an unnecessary prefix, the error "The image format is not supported" is reported.
url	No	String	Set either this parameter or **image**. Image URL. Currently, the following URLs are supported: ● Public HTTP/HTTPS URL ● URL provided by OBS. You need to be authorized to use OBS data, including service authorization, temporary authorization, and anonymous public authorization. For details, see **Configuring Access Permissions of OBS**. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the image to be detected resides is stable and reliable. OBS is recommended for storing image data.

Parameter	Mandatory	Type	Description
return_portrait_image	No	Boolean	Whether to return the portrait. The options are as follows: - **true**: The Base64 code of the portrait on the travel permit will be returned. - **false**: The Base64 code of the portrait on the travel permit will not be returned. If this parameter is not specified, **false** is used by default. In this case, the Base64 code of the portrait on the travel permit will not be returned.
return_portrait_location	No	Boolean	Whether to return the location of the portrait on the travel permit. The options are as follows: - **true**: The location of the portrait on the travel permit will be returned. - **false**: The location of the portrait on the travel permit will not be returned. If this parameter is not specified, **false** is used by default.

Response Parameters

Status code: 200

Table 4-118 Response body parameter

Parameter	Type	Description
result	**MainlandTravelPermitResult** object	Calling result of a successful API call This parameter is not included when the API fails to be called.

Table 4-119 MainlandTravelPermitResult

Parameter	Type	Description
name	String	Chinese name
name_en	String	Name in English
birth_date	String	Date of birth
sex	String	Gender
valid_period	String	Date of expiry
issuing_authority	String	Issuing authority
number	String	Permit number
issue_place	String	Place where the permit is issued
issue_times	String	Number of issuing times
id_name	String	Name on the back of the travel permit
id_number	String	Number on the back of the travel permit
machine_code1	String	Machine-readable code in the first line
machine_code2	String	Machine-readable code in the second line
machine_code3	String	Machine-readable code in the third line
type	String	Permit type. The options are as follows: ● **Mainland Travel Permit for Hong Kong and Macao Residents** ● **Mainland Travel Permit for Taiwan Residents**
side	String	Whether the front or back of the travel permit is displayed. The options are as follows: ● **front**: front side ● **back**: back side
portrait_image	String	Base64 code of the portrait. This parameter is returned only when **return_portrait_image** is set to **true**.

Parameter	Type	Description
portrait_location	Array<Array<Integer>>	Location of the portrait on the original image. This parameter is returned only when **return_portrait_location** is set to **true**. The image is displayed in a list. The list contains the two-dimensional coordinates (x,y) of the four vertices in the portrait area. The origin of the coordinates is the upper left corner of the image. The X axis is horizontal, and the Y axis is vertical.
confidence	**MainlandTravelPermitConfidence** object	Confidence of each field

Table 4-120 MainlandTravelPermitConfidence

Parameter	Type	Description
name	Float	Confidence of the Chinese name
name_en	Float	Confidence of the English name
birth_date	Float	Confidence of the birth date
sex	Float	Confidence of the gender
valid_period	Float	Confidence of the validity period
issuing_authority	Float	Confidence of the issuing authority
number	Float	Confidence of the permit number
issue_place	Float	Confidence of the place of issuance
issue_times	Float	Confidence of the number of issuing times
type	Float	Confidence of the permit type
side	Float	Confidence of information on the front and back sides of image of the permit
id_name	Float	Confidence of the name on the back of the travel permit
id_number	Float	Confidence of the number on the back of the travel permit
machine_code1	Float	Confidence of the machine-readable code in the first line
machine_code2	Float	Confidence of the machine-readable code in the second line

Parameter	Type	Description
machine_code3	Float	Confidence of the machine-readable code in the third line

Status code: 400

Table 4-121 Response body parameters

Parameter	Type	Description
error_code	String	Error code of a failed API call. For details, see **Error Codes**. This parameter is not included when the API is successfully called.
error_msg	String	Error message when the API call fails. This parameter is not included when the API is successfully called.

Example Request

- Example request (Method 1: Use the image Base64 string.)

```
POST  https://{endpoint}/v2/{project_id}/ocr/mainland-travel-permit
Request Header:
Content-Type: application/json,
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgA..."
    "return_portrait_image": True,
    "return_portrait_location": True
}
```

- Example request (Method 2: Use the image URL.)

```
POST  https://{endpoint}/v2/{project_id}/ocr/mainland-travel-permit
Request Header:
Content-Type: application/json,
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
    "url": "https://BucketName.obs.xxxx.com/ObjectName",
    "return_portrait_image": True,
    "return_portrait_location": True
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result" : {
    "name": "Zhang San",
    "name_en" : "ZHANG, SAN",
    "birth_date" : "1992.10.11",
    "sex": "M",
    "valid_period" : "1992.10.11-2002.04.04",
    "issuing_authority": "Bureau of Exit and Entry Administration of the Ministry of Public Security",
    "number" : "E23xxxxxx",
    "issue_times" : "00",
    "type": "Mainland Travel Permit for Hong Kong and Macao Residents",
    "side" : "front",
    "confidence" : {
      "name" : 0.9999,
      "name_en" : 0.9868,
      "birth_date" : 0.9984,
      "sex" : 0.9999,
      "valid_period" : 0.9988,
      "issuing_authority" : 0.9998,
      "number" : 0.9999,
      "issue_times" : 0.9997,
      "type" : 0.9999,
      "side" : 1.0
    }
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "AIS.0103",
  "error_msg" : "The image size does not meet the requirements."
}
```

Example SDK Code

The example SDK code is as follows:

Java

- Transfer the Base64 encoded string of the Chinese mainland travel permit image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMainlandTravelPermitSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);
        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMainlandTravelPermitRequest request = new 
RecognizeMainlandTravelPermitRequest();
        MainlandTravelPermitRequestBody body = new MainlandTravelPermitRequestBody();
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeMainlandTravelPermitResponse response = 
client.recognizeMainlandTravelPermit(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Transfer the URL of the Chinese mainland travel permit image for recognition.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.ocr.v1.region.OcrRegion;
import com.huaweicloud.sdk.ocr.v1.*;
import com.huaweicloud.sdk.ocr.v1.model.*;


public class RecognizeMainlandTravelPermitSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great 
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or 
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before 
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local 
environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        OcrClient client = OcrClient.newBuilder()
                .withCredential(auth)
                .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
                .build();
        RecognizeMainlandTravelPermitRequest request = new 
RecognizeMainlandTravelPermitRequest();
        MainlandTravelPermitRequestBody body = new MainlandTravelPermitRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeMainlandTravelPermitResponse response = 
client.recognizeMainlandTravelPermit(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Transfer the Base64 encoded string of the Chinese mainland travel permit image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMainlandTravelPermitRequest()
        request.body = MainlandTravelPermitRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_mainland_travel_permit(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Transfer the URL of the Chinese mainland travel permit image for recognition.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkocr.v1.region.ocr_region import OcrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkocr.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = OcrClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(OcrRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RecognizeMainlandTravelPermitRequest()
        request.body = MainlandTravelPermitRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_mainland_travel_permit(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Transfer the Base64 encoded string of the Chinese mainland travel permit image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMainlandTravelPermitRequest{}
    imageMainlandTravelPermitRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.MainlandTravelPermitRequestBody{
        Image: &imageMainlandTravelPermitRequestBody,
    }
    response, err := client.RecognizeMainlandTravelPermit(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Transfer the URL of the Chinese mainland travel permit image for recognition.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    ocr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/ocr/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := ocr.NewOcrClient(
        ocr.OcrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RecognizeMainlandTravelPermitRequest{}
    urlMainlandTravelPermitRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.MainlandTravelPermitRequestBody{
        Url: &urlMainlandTravelPermitRequestBody,
    }
    response, err := client.RecognizeMainlandTravelPermit(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More Programming Languages

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate corresponding SDK code examples.

Status Codes

Status Code	Description
200	Example response for a successful request
400	Example response for a failed request

See **Status Codes**.

Error Codes

See **Error Codes**.

5 Status Codes

An HTTP status code consists of three digits, which is classified into five categories: 1*xx*: related information; 2*xx*: operation successful; 3*xx*: redirection; 4*xx*: client error; 5*xx*: server error.

The following table lists the common status codes.

Status Code	Coding	Description
100	Continue	The server has received the initial part of the request and the client should continue to send the remaining part. It is issued on a provisional basis while request processing continues. It alerts the client to wait for a final response.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The target protocol must be more advanced than the source protocol. For example, the current HTTPS protocol is switched to a later version.
200	OK	The server has successfully processed the request.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The server has successfully processed the request, but is returning information that may be from another source.

Status Code	Coding	Description
204	No Content	The server has successfully processed the request, but does not return any content. The status code is returned in response to an HTTP OPTIONS request.
205	Reset Content	The server has successfully processed the request, but does not return any content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which the user or user agent (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. In such a case, there is no need to retransmit the resource since the client has a previously-downloaded copy.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	The HTTP status code is no longer used.
400	Bad Request	The request is invalid. The client should not repeat the request without modifications.
401	Unauthorized	The authorization information provided by the client is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	The server has received the request and understood it, but refuse to respond to it. The client should modify the request instead of re-initiating it.

Status Code	Coding	Description
404	Not Found	The requested resource cannot be found. The client should not repeat the request without modifications.
405	Method Not Allowed	The method specified in the request is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This status code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Timeout	The server has timed out waiting for the request. The client may repeat the request without modifications at a later time.
409	Conflict	The request could not be processed due to a conflict. For example, an edit conflict between multiple simultaneous updates or the resource that the client attempts to create already exists.
410	Gone	The requested resource has been deleted permanently and is no longer available. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to accept the request without a defined Content-Length.
412	Precondition Failed	The server did not meet one of the preconditions contained in the request.
413	Request Entity Too Large	The request is larger than the server is willing or able to process. The server may close the connection to prevent the client from continuing the request. If the server is only temporarily unable to process the request, the response will contain a Retry-After header field.
414	Request URI Too Long	The Request-URI is too long for the server to process.

Status Code	Coding	Description
415	Unsupported Media Type	The server is unable to process the media format in the request.
416	Requested Range Not Satisfiable	The requested range is invalid.
417	Expectation Failed	The server has failed to meet the requirements of the Expect request-header field.
422	Unprocessable Entity	The request is well-formed but cannot be processed due to semantic errors.
429	Too Many Requests	The client has sent an excessive number of requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server has received an excessive number of requests within a given time (beyond its processing capability). In this case, the client should resend the requests at any point after the time specified in the Retry-After header of the response.
500	Internal Server Error	The server is able to receive the request but unable to understand it.
501	Not Implemented	The server does not support the function required to fulfill the request.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server.
503	Service Unavailable	The requested service is invalid. The client should not repeat the request without modifications.
504	Gateway Timeout	The request cannot be fulfilled within a given time. This status code is returned to the client only if the **Timeout** parameter is specified in the request.
505	HTTP Version Not Supported	The server does not support the HTTPS protocol version used in the request.

6 Error Codes

If an error occurs during API calling, no result will be returned. You can locate the cause of an error using the error code of each API. When an API call fails, HTTPS status code 4*xx* or 5*xx* is returned. The returned message body contains a specific error code and error message. If you fail to identify the cause of an error, contact Huawei Cloudcustomer service technical support and provide the error code for quick troubleshooting.

Format of an Error Response Body

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_msg": "The input parameter is invalid.",
    "error_code": "AIS.0101"
}
```

In the response body, **error_code** indicates the error code, and **error_msg** provides information about the error.

Error Codes

OCR is deployed on ModelArts and uses API Gateway (APIG). Therefore, OCR error codes may start with ModelArts, APIGW, or APIG.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to **Error Codes**. If an error code starts with APIG, rectify the fault by referring to this table.

Module	Error Code	Error Message	Description	Solution
OCR	AIS.0101	The input parameter is invalid.	The input parameter does not meet the requirements.	Check whether the input parameter is correct. Possible causes are as follows: ● **image** and **url** are both configured or the value format is incorrect. ● **true** or **false** is incorrectly spelled. ● Incorrect characters are included in the parameter value. For details about the parameter value format, see the request parameter description of each API.
	AIS.0102	The image format is not supported.	The image format is not supported, or the value of **image** is not a Base64 encoded string.	Check the image format. For details about the image formats supported by each service, see **Constraints and Limitations**.
	AIS.0103	The image size does not meet the requirements.	The image size does not meet the requirements.	Check the image size. For details about the image size supported by each service, see **Constraints and Limitations**.
	AIS.0104	The image is not supported or the image quality is poor.	The image is not supported or is of poor quality.	Check the image type and quality. Check whether the called API is correct.
	AIS.0105	Recognition failed.	Algorithm calculation failed.	Contact **Huawei Cloud support personnel**.

Module	Error Code	Error Message	Description	Solution
	AIS.0117	Input ID does not exist.	The entered ID does not exist.	Check whether **classifier_id** or **template_id** is specified.
	AIS.0118	Service request error or the tax authority's interface was malfunctioning.	The tax agency's interface timed out and did not respond, resulting in an error in the service request.	Wait for the tax agency's interface to recover. (You are advised to try again 2 hours later.)
	AIS.0119	Template match failed, template_id={xxx}.	The input image failed to match the template.	Check whether the reference fields in the template match the fields in the input image.
	AIS.0120	Classification failed.	The image category failed to be entered.	Check whether the template in the classifier matches the input image.
	AIS.0121	The recognition area of template is not configured, template_id={xxx}	No recognition area is configured for the template.	Check whether the recognition area is configured for the template.
	AIS.0122	The image contains two or more ID cards with the same side.	There are two or more ID cards in the image that are on the same side.	Check if there are multiple ID cards on the same side in the input image. If the **side** parameter of ID Card OCR is **double_side**, the image cannot contain two or more ID cards on the same side.
Platform	ModelArts.0203	Invalid token.	The token is invalid.	Check whether the token is correct.

Module	Error Code	Error Message	Description	Solution
	ModelArts.4101	Token header cannot be empty.	The token is empty.	The HTTP request header does not contain the token request authentication information of **x-auth-token**. Check the request.
	ModelArts.4102	Parse Token error.	The token failed to be parsed.	The token request authentication information of **x-auth-token** in the HTTP request header is incorrect. Check the sent request and token.
	ModelArts.4103	Invalid Token header.	The token is invalid.	The token request authentication information of **x-auth-token** in the HTTP request header is incorrect. Check the sent request and token.
	ModelArts.4104	Invalid Request Content Length.	The length of the request body is invalid.	Check the request body length. The size cannot exceed 10 MB.
	ModelArts.4105	The JSON format of the input data is incorrect.	The JSON format of the request body is incorrect.	Check the JSON format of the request body.
		Failed to obtain the temporary AK/SK	The temporary AK/SK failed to be obtained.	Contact technical support. (If OBS is used, check the authorization first.)
	ModelArts.4106	Invalid authorization request.	The account is restricted.	Check the user's resources. For details about the account restriction reason, see **My Account FAQ** in **Help Center**.

Module	Error Code	Error Message	Description	Solution
	ModelArts.4107	Get user temp ak sk error.	An exception occurred when obtaining the temporary AK/SK.	Contact **Huawei Cloud support personnel**.
	ModelArts.4201	Request url need service id.	The request URL does not contain the service ID.	Check the service ID in the request URL.
	ModelArts.4202	Request url format invalid.	The request URL format is invalid.	Check the request URL format.
	ModelArts.4203	Access denied! You do not have permission.	Access permission is unavailable.	Check the access permission.
	ModelArts.4204	Request api error! The API XXX is not subscribed.	Failed to request the service because the service has not been subscribed.	● Go to the **OCR** console, select an appropriate region, and subscribe to the API you need. ● If the service has been subscribed to, check whether the region (or account) where the service is subscribed to is the same as the region (or account) where the service is called. If they are the same, check whether the URL of the API is spelled correctly.
	ModelArts.4301	Have error when get ai-service admin token.	Failed to obtain the admin token of the service.	Contact Huawei Cloud support personnel.
	ModelArts.4302	Gateway forwarding error.	The serviced failed to be accessed.	Contact Huawei Cloud support personnel.

Module	Error Code	Error Message	Description	Solution
	ModelArts.4401	Max concurrency error.	The number of concurrent requests exceeds the upper limit.	Contact **Huawei Cloud support personnel**.
	ModelArts.4402	Backend service timeout error.	Service processing timed out.	Contact **Huawei Cloud support personnel**.
	ModelArts.4403	Backend service unavailable error.	The service is unavailable because it failed to respond or its failure rate is high.	Contact **Huawei Cloud support personnel**.
	ModelArts.4502	Backend service response error.	The service failed to respond.	Contact **Huawei Cloud support personnel**.
	ModelArts.4503	Backend service found error.	The service does not exist.	Contact **Huawei Cloud support personnel**.
	ModelArts.4504	Backend service api not found.	The API to be accessed does not exist.	Contact **Huawei Cloud support personnel**.
	ModelArts.4505	Backend service internal error.	The internal service error occurs.	Contact **Huawei Cloud support personnel**.
	ModelArts.4506	SubService service found error.	The subservice corresponding to the API does not exist.	Contact **Huawei Cloud support personnel**.
	ModelArts.4508	Service not start.	The service does not exist or is stopped.	Contact **Huawei Cloud support personnel**.
	ModelArts.4601	The URL is not allowed.	The external URL is invalid.	Check the format of the entered download address.
	ModelArts.4603	Obtaining the file from the URL failed.	The file failed to be downloaded from the external URL.	Check the network and URL.

Module	Error Code	Error Message	Description	Solution
	ModelArts.4702	Query Obs agency failed.	The OBS agency failed to be queried.	Check whether the OBS agency has been enabled for the service.
	ModelArts.4703	The Obs URL is invalid.	The OBS URL is invalid.	Check the OBS URL.
	ModelArts.4704	Obtaining the file from the OBS failed.	The OBS file failed to be obtained.	Check the OBS file.
	ModelArts.4705	The file stored on the OBS is oversized.	The OBS file is oversized.	Check the size of the OBS file and ensure that the file does not exceed the size limit.
	ModelArts.4706	The Obs file is not exist.	The OBS file does not exist.	Check whether the corresponding file exists.
APIG	APIG.0101	The API does not exist or has not been published in the environment.	The API does not exist or has not been published.	● Check whether the API URL is correct. For example, check whether the project ID is included in the URL. ● Check whether the region in the URI is the same as that configured for calling the service by referring to **Endpoints**. ● Check whether the HTTP request method (such as POST and GET) is correct. ● For details about the URI, see the corresponding API page.

Module	Error Code	Error Message	Description	Solution
	APIG.0201	Backend timeout.	Request timed out.	- Check whether the original API call requests are excessively frequent. If so, use the retry mechanism to rectify the fault by checking the return value in the code and retrying the requests after a short period of time (for example, 2 to 5 seconds). Alternatively, check whether the result of the previous request is returned at the backend. If it is, send the next request. This helps prevent excessively frequent requests. - Check whether the image is too large or the network delay is too long. If the image is too large, compress the image in proportion while ensuring the image definition. If the network delay is long, you can increase the network transmission speed.

Module	Error Code	Error Message	Description	Solution
	APIG.0301	Incorrect IAM authentication information.	The IAM authentication information is incorrect. - **decrypt token fail**: The token fails to be parsed. - **token expires**: The token expires. - **verify aksk signature fail**: The AK/SK authentication fails. - **x-auth-token not found**: The **x-auth-token** parameter is not found.	- If the token fails to be parsed, check the method for obtaining the token, whether the request body is correct, whether the token is correct, and whether the environment for obtaining the token is the same as the environment for calling the token. - If the token expires, obtain a new token that is valid permanently. - Check whether the AK/SK pair is correct. For example, the SK corresponding to the AK is incorrect; an extra space is entered in the AK/SK pair. - AK/SK-based authentication errors occur frequently. If an AK/SK pair fails to be authenticated for more than five consecutive times, the AK/SK pair is locked for 5 minutes (the AK/SK-based authentication is considered as an abnormal

Module	Error Code	Error Message	Description	Solution
				authentication request within 5 minutes). After 5 minutes, the AK/SK pair is unlocked and re-authenticated. ● Check the account permissions, for example, whether the account is in arrears or frozen. ● Check whether the spelling of **X-Auth-Token** in the request header is correct.
	APIG.0308	The throttling threshold has been reached: policy user over ratelimit,limit:XX,time:1 minute.	The request exceeds the default rate limit of the service.	Rectify the fault by following the instructions provided in **Why Is a Message Stating "APIG.0308" Displayed When the OCR API Is Called?**
Other	If other error codes are displayed, **contact us**.			

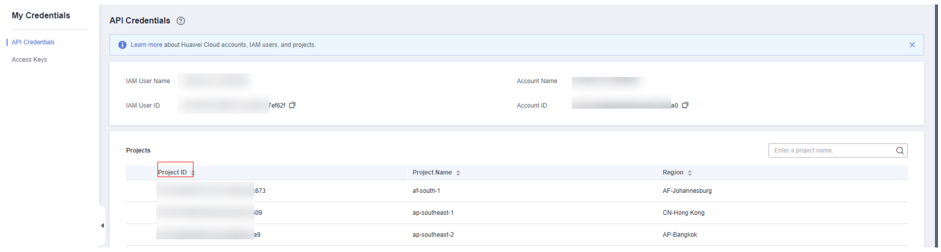
7 Appendix

7.1 Obtaining a Project ID

Obtaining a Project ID from the Management Console

1. Log in to the management console.
2. Hover the cursor over your username in the upper right corner and click **My Credentials** from the drop-down list.
3. On the **My Credentials** > **API Credentials** page, view the username and account name and view projects in the project list.

Figure 7-1 Viewing the project ID



IAM User Name [redacted]
IAM User ID [redacted]7ef62f
Account Name [redacted]
Account ID [redacted]a0

Project ID	Project Name	Region
[redacted]:673	af-south-1	AF-Johannesburg
[redacted]509	ap-southeast-1	CN-Hong Kong
[redacted]a9	ap-southeast-2	AP-Bangkok

If there are multiple projects in one region, expand **Region** and view subproject IDs in the **Project ID** column.

Obtaining a Project ID by Calling an API

The API for obtaining a project ID is **GET https://{endpoint}/v3/projects**. **{Endpoint}** indicates the endpoint of IAM. For details about API authentication, see **Authentication**.

For example, if OCR is deployed in the **ap-southeast-1** region, **name** is **ap-southeast-1**,, and **id** in **projects** is the project ID. If an error is reported when obtaining the project ID, check whether *{endpoint}* is correct.

```
GET https://iam.ap-southeast-1.myhuaweicloud.com/v3/projects
{
   "projects": [
      {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "ap-southeast-1",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://support-intl.huaweicloud.com/zh-cn/devg-apisign/api-sign-
provide.htmlcd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

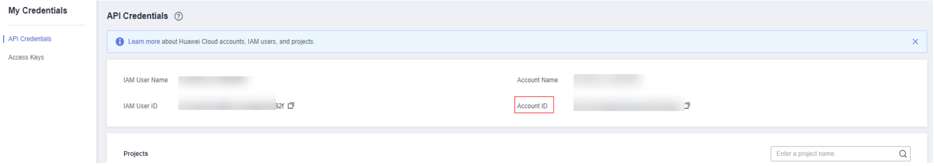
7.2 Obtaining an Account ID

An account ID (domain-id) is required for some URLs when an API is called. To obtain the ID, perform the following operations:

1. Log in to the management console after registration.
2. Hover the cursor over your username and choose **My Credentials**.

   On the **My Credentials** page, view **Account ID**.

   Figure 7-2 Viewing the account ID

   My Credentials

   API Credentials

   Access Keys

   API Credentials

   Learn more about Huawei Cloud accounts, IAM users, and projects.

   IAM User Name [redacted]

   IAM User ID [redacted]62f

   Account Name [redacted]

   Account ID [redacted]

   Projects

   Enter a project name.

7.3 Configuring Access Permissions of OBS

Multimedia files such as images and audio files in the Enterprise Intelligence (EI) services can be directly processed by OBS. This reduces service usage costs, shortens service response time, and improves service experience.

To ensure data security, a service can used authorized URLs (**https://<bucket-name>.<endpoint>/<object-name>**) to access files stored on OBS after it is granted with the required permission. Unauthorized services cannot obtain user data. To obtain the user data in this case, enable public read authorization or provide a URL that has been temporarily authorized.

Enabling Authorization for OCR

To use data in OBS, you need to enable OBS authorization. Go to **the OCR console**. Enable **OBS Authorization**. After the authorization is enabled, you can use the authorized URL to access the service.

Figure 7-3 OBS authorization

If you want to use data stored on OBS, you can authorize OCR to access your OBS.

NOTE

OBS must reside in the same region as OCR.

For details about the constraints on using OBS features, see **Restrictions and Limitations**.

(Optional) Enabling Public Read Authorization

To do so, configure the bucket policy to **Public Read**. For details, see **Configuring a Standard Bucket Policy** in *Object Storage Service Console Operation Guide*. This method is not recommended for private data.

Using Temporarily Authorized Requests for Authentication

Public read authorization is easy to use. However, when it is enabled, sensitive information, such as private data, may be disclosed. In this scenario, the temporary authorization function provided by OBS comes in handy.

OBS allows users to construct a specific URL for objects in OBS. The URL contains authentication information. Any user can use the URL to access the specified object in OBS, but the URL is valid only before the expiry time specified in **Expires**. After a user issues temporary authorization, other users can perform operations without requiring the user's access key.

For details about how to use the OBS temporary authorization function, see section "Authorized Access" in the **Object Storage Service SDK Reference**. Download the related SDK and sample code, and compile code to obtain the related URL.

A Change History

Release Date	Description
2022-11	APIs deployed in the **CN-Hong Kong**, **AP-Bangkok**, and **AP-Singapore** regions support enterprise project management. Financial statistics can be collected by enterprise project.
2022-07	Added the following OCR API: **ID Document**
2022-03	Added the following OCR API: **Cambodian ID Card**
2022-01	Added the following OCR APIs: **Hong Kong ID Card** **Macao Resident ID Card** **Exit-Entry Travel Permit** **Chinese Mainland Travel Permit**
2020-03	Added the following OCR API: **Thailand Plate Number**
2020-11	Added the following OCR API: **Chile ID Card**
2019-07	Added the following OCR API: **Myanmar Driving License**
2019-04	This issue is the first official release.