

文字识别服务

API 参考

文档版本 01
发布日期 2025-08-20



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

1 使用前必读.....	1
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束与限制.....	2
1.5 基本概念.....	2
2 API 概览.....	4
3 如何调用 API.....	6
3.1 总览.....	6
3.2 开通服务.....	7
3.3 构造请求.....	10
3.4 认证鉴权.....	12
3.5 返回结果.....	15
4 API.....	17
4.1 智能文档解析.....	17
4.2 通用证件识别.....	34
4.3 智能证件分类.....	46
4.4 通用表格识别.....	58
4.5 通用文字识别.....	70
4.6 网络图片识别.....	83
4.7 护照识别.....	94
4.8 泰文身份证识别.....	105
4.9 哥伦比亚身份证识别.....	120
4.10 柬文身份证识别.....	129
4.11 缅文身份证识别.....	143
4.12 缅文驾驶证识别.....	156
4.13 智利身份证识别.....	168
4.14 越南身份证识别.....	179
4.15 秘鲁身份证识别.....	189
4.16 泰国车牌识别.....	202
5 状态码.....	213

6 错误码.....	216
7 附录.....	223
7.1 获取项目 ID.....	223
7.2 获取账号 ID.....	224
7.3 配置 OBS 访问权限.....	224

1 使用前必读

1.1 概述

OCR以开放API（Application Programming Interface，应用程序编程接口）的方式提供给用户，用户通过实时访问和调用API获取推理结果，帮助用户自动采集关键数据，打造智能化业务系统，提升业务效率。调用API时，需要用户网络可以访问公网。

您可以使用本文档提供文字识别服务API的描述、语法、参数说明及样例等内容，进行相关操作，例如文字识别包含的通用类、证件类和票据类等具体接口使用说明。不同区域所部署的接口不同，详情请参见[终端节点](#)；接口所支持的全部操作请参见[2 API概览](#)。

在调用文字识别API之前，请确保已经充分了解文字识别相关概念。

OCR还提供多种编程语言的SDK供您使用，SDK使用方法请参考《[SDK参考](#)》。

1.2 调用说明

文字识别提供了REST（Representational State Transfer）风格的API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

调用API时，需要用户网络可以访问公网。

同时文字识别还提供多种编程语言的SDK供您使用，SDK的使用方法请参见《[SDK参考](#)》。

1.3 终端节点

终端节点即调用API的请求地址，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询所有服务的终端节点。

文字识别的终端节点如[表1-1](#)所示，请您根据业务需要选择对应区域的终端节点。

业务的服务器部署区域与OCR服务区域不一致，只要服务器可以访问公网，即可使用OCR服务。一般情况下，建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。

表 1-1 文字识别的终端节点

区域名称	区域	终端节点 (Endpoint)	部署服务
中国-香港	ap-southeast-1	ocr.ap-southeast-1.myhuaweicloud.com ocr.ap-southeast-1.myhuaweicloud.cn	护照识别、缅文身份证识别、缅文驾驶证识别、通用文字识别、越南身份证、智能文档解析
亚太-曼谷	ap-southeast-2	ocr.ap-southeast-2.myhuaweicloud.com ocr.ap-southeast-2.myhuaweicloud.cn	泰文身份证识别、护照识别、网络图片识别、泰国车牌识别、缅文身份证
拉美-圣地亚哥	la-south-2	ocr.la-south-2.myhuaweicloud.com ocr.la-south-2.myhuaweicloud.cn	智利身份证识别、秘鲁身份证
亚太-新加坡	ap-southeast-3	ocr.ap-southeast-3.myhuaweicloud.com ocr.ap-southeast-3.myhuaweicloud.cn	柬文身份证识别、通用表格识别、智能文档解析
拉美-墨西哥城二	la-north-2	ocr.la-north-2.myhuaweicloud.com	哥伦比亚身份证识别

1.4 约束与限制

更详细的限制请参见具体API的说明和[产品介绍约束与限制](#)。

1.5 基本概念

- 账号
用户注册华为云时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常管理工作。
- 用户

由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

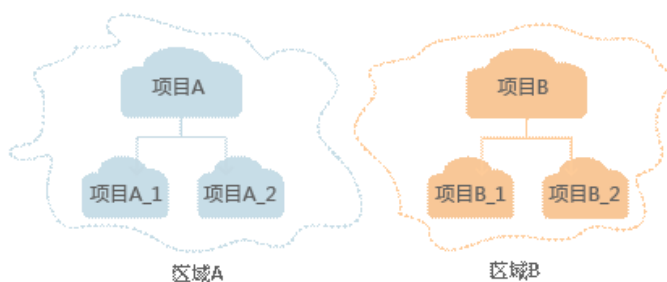
- 可用区（AZ，Availability Zone）

一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 项目

华为云的区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中购买资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



2 API 概览

通过使用文字识别服务的自研API，您可以使用文字识别服务的所有功能，如表2-1所示。

表 2-1 接口说明

API	说明	部署区域
智能文档解析	对证件、票据、表单等任意版式文档进行键值对提取、文字识别、以及表格识别等任务，实现进阶高效的自动化结构化返回。	中国-香港(ap-southeast-1) 亚太-新加坡 (ap-southeast-3)
智能证件分类	支持9类证件的分类和告警检测，以JSON格式返回结果。支持的证件类型有秘鲁身份证、柬文身份证、香港身份证、澳门身份证、缅文身份证、缅文驾驶证、泰文身份证、护照和中华人民共和国居民身份证。	亚太-曼谷(ap-southeast-2)
通用表格识别	识别表格图片中的文字内容，并将识别的结果以JSON格式返回给用户。返回结果将图片区域（ words_region ）划分为两类：纯文本区（ text ）和表格区（ table ），并返回表格结构（ row, column ）和文本信息。该接口的使用限制请参见 约束与限制 ，详细使用指导请参见 OCR服务使用简介 章节。	亚太-新加坡 (ap-southeast-3)
通用文字识别	识别图片上的文字信息，以JSON格式返回识别的文字和坐标。支持扫描文件、电子文档、书籍、票据和表单等多种场景的文字识别。 支持中英文以及部分繁体字。该接口的使用限制请参见 约束与限制 ，详细使用指导请参见 OCR服务使用简介 章节。	中国-香港(ap-southeast-1)

API	说明	部署区域
护照识别	识别护照首页图片中的文字信息，并以JSON格式返回识别的结构化结果。	中国-香港(ap-southeast-1) 亚太-曼谷(ap-southeast-2)
网络图片识别	识别网络图片中的文字内容，并以JSON格式返回识别的结构化结果。	亚太-曼谷(ap-southeast-2)
泰文身份证识别	识别泰国身份证中的文字信息，并返回识别的结构化结果。	亚太-曼谷(ap-southeast-2)
缅文身份证识别	识别缅甸身份证中的文字信息，并返回识别的结构化结果。	中国-香港(ap-southeast-1)
柬文身份证识别	识别柬文身份证图片中的文字内容，并将识别的结构化结果返回给用户。	亚太-新加坡(ap-southeast-3)
缅文驾驶证识别	识别缅甸驾驶证中的文字信息，并以JSON格式返回识别的结构化结果。	中国-香港(ap-southeast-1)
智利身份证识别	识别智利身份证图片中的文字内容，并将识别的结果以JSON格式返回给用户。	拉美-圣地亚哥(la-south-2)
越南身份证识别	识别越南身份证中的文字信息，并将识别的结构化结果返回给用户。	中国-香港(ap-southeast-1)
秘鲁身份证识别	识别秘鲁身份证图片中的文字内容，并将识别的结构化结果返回给用户。	拉美-圣地亚哥(la-south-2)
泰国车牌识别	识别泰国车牌图片中的车牌信息，返回识别结果和车牌的区域位置信息。	亚太-曼谷(ap-southeast-2)
通用证件识别	识别身份证件图像，并将识别的结构化结果返回给用户。支持多个国家/地区的身份证、驾驶证和护照。	中国-香港(ap-southeast-1)

3 如何调用 API

3.1 总览

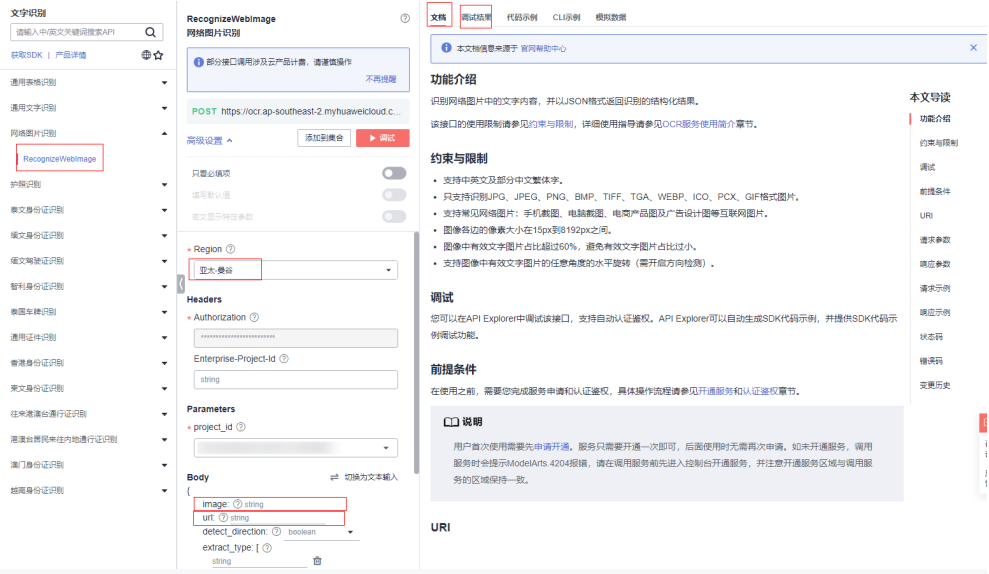
调用API可以通过Postman或SDK进行调试。包含开通服务、配置环境、认证鉴权和构造请求（调用服务）。

1. 开通服务：包含注册登录、创建IAM账号（可选）、开通服务，详情请参考[开通服务](#)章节。
2. 配置环境：可选择[下载Postman](#)或[下载SDK](#)。
3. 认证鉴权：可选择[Token认证](#)（Postman环境）或[AK/SK认证](#)（SDK环境）。根据步骤二配置的环境选择认证鉴权的方式。
4. 构造请求：Postman环境的构造请求请参考[构造请求](#)章节，SDK环境的API调用详见[SDK参考](#)。

除上述两种方式外，还可通过[APIE在线调试](#)的方式调用API，步骤如下。

1. [开通服务](#)后进入APIE界面。
2. 选择已开通的服务和开通服务对应的Region。
3. Body部分填入image的base64编码或url均可，其它参数根据实际情况选填，参数部分可参考APIE界面的“接口文档”。
4. 单击调试，在“调试结果”中查看调用结果。

图 3-1 APIE 在线调试



3.2 开通服务

- 步骤1 在[华为云官网](#)注册华为云账号，并登录。账号不能处于欠费或冻结状态。
- 步骤2 创建IAM账号（可选）：

1. 登录华为云[控制台](#)，在右上角用户名处选择“统一身份认证”。

图 3-2 统一身份认证



2. 单击“创建用户”，并设置IAM账户的用户名、密码，创建IAM用户。

图 3-3 创建用户

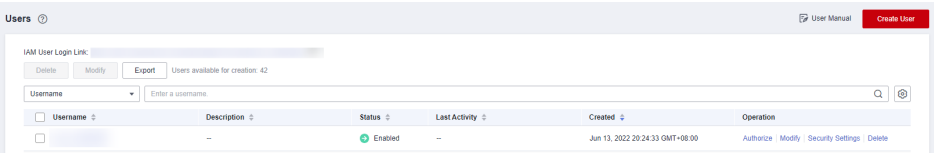
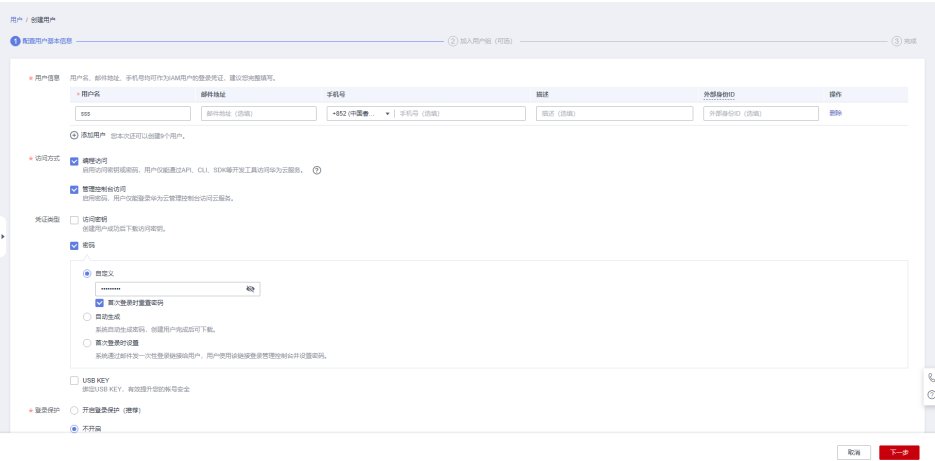


图 3-4 设置用户名、密码



3. 给予账户授予权限请参考[权限管理](#)。

图 3-5 设置用户组



步骤3 开通服务：

1. 单击“控制台”>“服务列表”，找到“文字识别OCR”，进入[文字识别控制台](#)。

图 3-6 文字识别控制台



2. 在文字识别控制台“总览”页面，选择对应产品的“终端节点”。不同区域OCR产品部署情况请参考[终端节点](#)。

图 3-7 终端节点



3. 选择产品类别，如：通用类、证件类等，然后选择需要使用的服务，执行开通操作。计费方式为按需付费。

图 3-8 服务开通

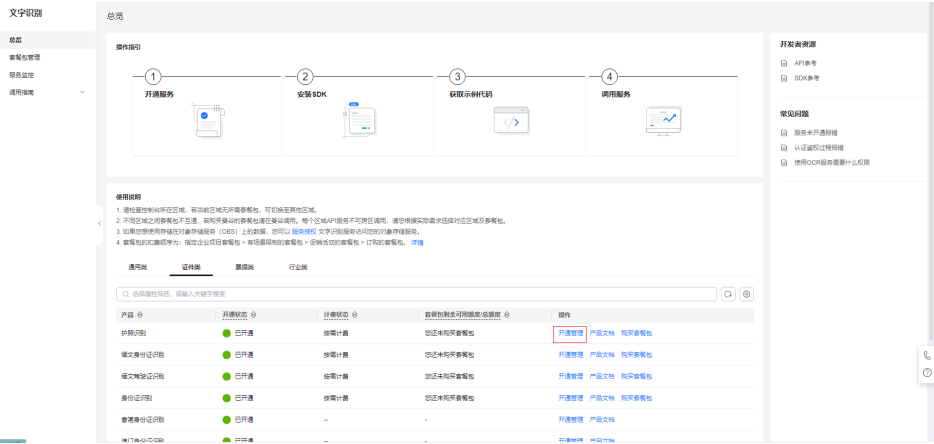
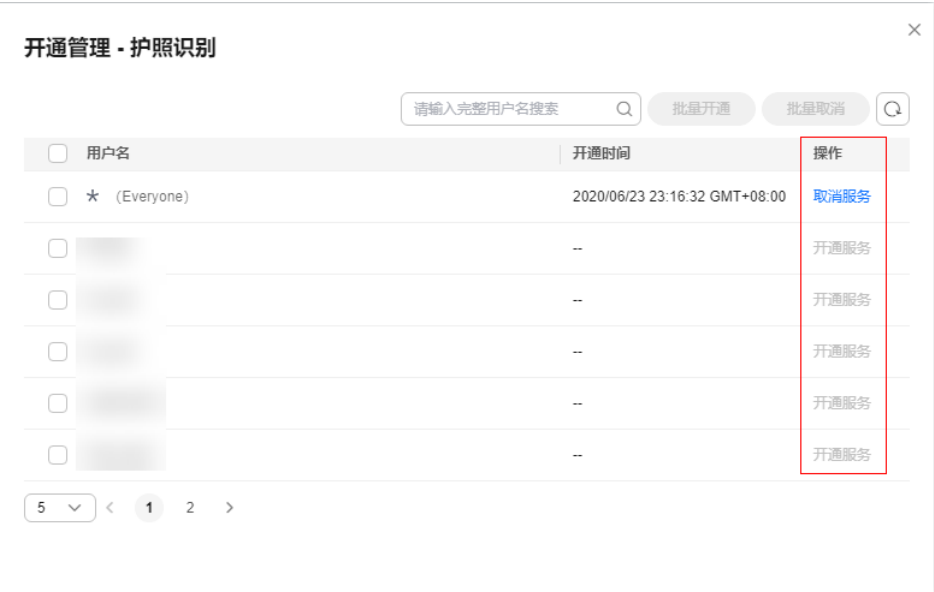


图 3-9 开通管理



说明

- 若主账号授予某个IAM子账号所有权限（OCR FullAccess），则主账号或该IAM子账号均可开通OCR服务。未被授予所有权限的子账号不可自行开通服务。

若主账号授予IAM子账号只读访问权限（OCR ReadOnlyAccess）或不授予任何权限，该IAM子账号若需开通OCR服务，则需要主账号（或者被授予所有权限的子账号）进行开通管理。
- 或者直接单击购买套餐包也可开通服务。计费方式为套餐包付费。

图 3-10 购买套餐包



4. 服务开通成功后，开通状态显示“已开通”。

图 3-11 已开通



说明

- OCR开通服务时，计费规则默认为“按需计费”，按需计费时，不使用OCR服务，则不收费。如果您购买了套餐包，套餐包扣减规则请参见[计费说明](#)，套餐包购买后暂不支持退款。
- 如未开通服务，直接调用OCR API会提示ModelArts.4204报错。
- 使用文字识别服务时，可以通过配置OBS访问权限搭配使用。服务只需要配置一次即可，后面使用时无需再次配置，详细信息请参见[配置OBS访问权限](#)章节。

----结束

3.3 构造请求

本节介绍REST API请求的组成，说明如何调用API。

请求 URI

请求URI由如下部分组成。

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

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

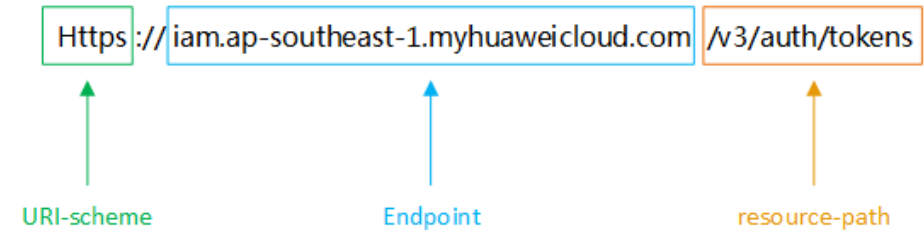
表 3-1 URI 中的参数说明

参数	说明
URI-scheme	表示用于传输请求的协议，当前所有API均采用HTTPS协议。
endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的endpoint不同，您可以从 终端节点 章节查看。
resource-path	资源路径，也即API访问路径。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如，在“亚太-曼谷”区域调用API，则需使用“亚太-曼谷”区域的endpoint（ ocr.ap-southeast-2.myhuaweicloud.com ），并在[获取用户Token](#)的URI部分找到 resource-path（ /v3/auth/tokens ），拼接起来如下所示。

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

图 3-12 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
GET	请求服务器返回指定资源。
PUT	请求服务器更新指定资源。

方法	说明
POST	请求服务器新增资源或执行特殊操作。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	请求服务器资源头部。
PATCH	请求服务器更新资源的部分内容。 当资源不存在的时候，PATCH可能会去创建一个新的资源。

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

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

- **Content-Type**: 消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **X-Auth-Token**: 用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token请参考认证鉴权章节。

添加消息头后的请求如下所示。

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

请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示。

```
POST https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZlhcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAABwESAAMAA..."
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用curl、Postman或直接编写代码等方式发送请求调用API。

3.4 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证调用请求。
- AK/SK认证：通过AK (Access Key ID) /SK (Secret Access Key)加密调用请求。

Token 认证

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

说明

- Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。
- 如果您的华为云账号已升级为华为账号，将不支持获取账号Token。建议您为自己创建一个IAM用户，获取IAM用户的Token。

Token可通过调用“获取用户Token”接口获取，调用本服务API需要project级别的Token，即调用获取用户Token接口时，请求body中“auth.scope”的取值需要选择“project”，请求示例如下所示。

username、**domainname**、**project name**可登录控制台“[我的凭证 > API凭证](#)”页面获取。**password**为用户密码。

- 伪码

POST https://iam.ap-southeast-2.myhuaweicloud.com/v3/auth/tokens //以获取亚太-曼谷区域Token为例
Content-Type: application/json

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", //IAM用户名
          "password": "*****", //密码
          "domain": {
            "name": "domainname" //账号名
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "project name" //替换为实际的project name，如ap-southeast-1
      }
    }
  }
}
```

- Java

```
package xxx; // GetToken类所在的项目路径

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 //填写正确的账号名、IAM用户名、密码
= "{\"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"])
```

如下图所示，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。获取Token之后，您就可以使用Token认证调用OCR服务API。

图 3-13 获取用户 Token 响应消息头

```
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 → MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA...
x-subject-token → MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA...
```

```
x-xss-protection → 1; mode=block;
```

AK/SK 认证

说明

AK/SK 签名认证方式仅支持消息体大小 12M 以内，12M 以上的请求请使用 Token 认证。

AK/SK 认证就是使用 AK/SK 对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK (Access Key ID)：访问密钥 ID。与私有访问密钥关联的唯一标识符；访问密钥 ID 和私有访问密钥一起使用，对请求进行加密签名。
- SK (Secret Access Key)：与访问密钥 ID 结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用 AK/SK 认证时，您可以基于签名算法使用 AK/SK 对请求进行签名，也可以使用专门的签名 SDK 对请求进行签名。详细的签名方法和 SDK 使用方法请参见 [API 签名指南](#)。

如果之前没有生成过 AK/SK，可登录“[我的凭证](#)”界面，选择“管理访问密钥 > 新增访问密钥”来获取。

须知

签名 SDK 只提供签名功能，与服务提供的 SDK 不同，使用时请注意。

AK/SK 获取方式请参考 [获取 AK/SK](#)。

3.5 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组数字代码，状态码表示了请求响应的状态，如果调用后返回状态码为 2xx，则表示调用成功。完整的状态码列表请参见[状态码](#)。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。例如[表3-3](#)所示。

表 3-3 响应消息头

名称	说明
Content-Length	响应消息体的字节长度，单位为Byte。
Date	系统响应的时间。
Content-Type	响应消息体的MIME类型。

响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

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

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 智能文档解析

功能介绍

对证件、票据、表单等多版式文档进行文字识别、版面分析、键值对提取以及表格识别等任务，并以json格式返回识别的结构化结果。

约束与限制

- 支持中英文及部分中文繁体字。
- 只支持识别PNG、JPG、JPEG、BMP、GIF、TIFF、WEBP、PCX、ICO、PSD格式的图片以及PDF文档，PDF只支持单页识别，可通过pdf_page_number参数指定页码。
- 图像各边的像素大小在15px到30000px之间。单个图片、PDF文件其对应的Base64编码不超过10MB。
- 图像中识别区域有效占比超过80%，保证所有文字及其边缘包含在图像内。
- 支持图像任意角度的水平旋转。
- 单页字符数不大于1800，以获得较优的识别效果。
- 目前不支持复杂背景（如户外自然场景、防伪水印等）和文字扭曲图像的文档解析。
- 文字识别服务属于公有云服务，线上用户资源共享，如果需要多并发请求，请提前[联系我们](#)。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

POST /v2/{project_id}/ocr/smart-document-recognizer

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限，响应消息头中X-Subject-Token的值即为Token。

参数	是否必选	参数类型	描述
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。获取方法：进入“企业项目管理”页面，单击企业项目名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 • 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 • 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 • 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
data	否	String	与url二选一。单个图片、PDF文件其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议文件大小不超过7MB。图像尺寸不小于15×15像素，最长边不超过30000像素，支持JPG/PNG/BMP/TIFF格式。PDF以150dpi的分辨率转为图像进行文档解析，需符合上述图像尺寸规定。若PDF有多页，当前仅对第1页进行识别。

参数	是否必选	参数类型	描述
url	否	String	与data二选一。url中单个图片、PDF文件其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议文件大小不超过7MB。目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 • url中不能存在中文字符，若存在，中文需要进行utf8编码。
single_orientation_mode	否	Boolean	单朝向模式开关。可选值包括： • true：打开单朝向模式 • false：关闭单朝向模式 图片文字方向一致时，打开该开关可提升识别精度；图片文字方向不一致时，关闭该开关可支持多朝向文字识别。未传入该参数时默认为true，既默认图片中的字段为单朝向。

参数	是否必选	参数类型	描述
language	否	String	语种选择，未传入该参数时默认为中英文识别模式。 - zh: 中英文 - es: 西班牙语 - ar: 阿拉伯语 - pt: 葡萄牙语 - tr: 土耳其语 - th: 泰语 说明 - 表格、表单识别和版面分析支持西班牙语、阿拉伯语、葡萄牙语、土耳其语和泰语识别。 - 键值对提取能力（KV）仅支持中英文模式，暂不支持其他语种。
kv	否	Boolean	是否进行键值对（key-value）提取。若是，结果会以“kv_result”这一关键字返回。
table	否	Boolean	是否进行表格识别。此处表格特指逻辑表格，通常具有M行N列的形式，且第一行或第一列为表头。若是，结果会以“table_result”这一关键字返回。
layout	否	Boolean	是否进行版面分析。若是，结果会以“layout_result”这一关键字返回。
return_excel	否	Boolean	仅当table为True时有效。是否返回表格转换Microsoft Excel的Base64编码字段。
form	否	Boolean	是否进行有线表单识别。有线表单指关键信息以有线单元格形式进行呈现，例如户口本、机动车发票等。若是，结果会以“form_result”这一关键字返回。

参数	是否必选	参数类型	描述
formula	否	Boolean	是否进行公式识别，识别结果为 latex 序列。若是，结果会以 “formula_result” 这一关键字返回。 - 开启公式识别后会降低响应速度。 - 当前仅支持3行以内公式识别，不支持3行以上的多行公式。
kv_map	否	String	需要传入字典的 json 序列化后字符串，用于对 kv_result 中的特定 key 值进行归一化映射。例如，kv_result 中包含 {“名称”: “小明”} 的键值对，若传入 {“名称”: “姓名”} 的 kv_map，则返回结果为 {“姓名”: “小明”}。 说明 参数传入示例： “kv_map”: {“名称”: “姓名”}
erase_seal	否	Boolean	是否进行印章擦除。开启后，可提升印章遮挡区域的文字识别精度。
pdf_page_number	否	Integer	指定 PDF 页码识别。传入该参数时，则识别指定页码的内容。如果不传该参数，则默认识别第 1 页。

响应参数

状态码： 200

表 4-4 响应 Body 参数

参数	参数类型	描述
result	Array of SmartDocumentRecognizerResult objects	调用成功时返回的结果列表，按页面顺序返回，列表第一项为第一页识别结果，依次类推。调用失败时无此字段。

表 4-5 SmartDocumentRecognizerResult

参数	参数类型	描述
ocr_result	SmartDocumentRecognizerOcrResult object	文字识别结果。
kv_result	SmartDocumentRecognizerKvResult object	键值对提取结果。当输入参数“kv”为“true”时，才返回该参数。
table_result	SmartDocumentRecognizerTableResult object	表格识别结果。当输入参数“table”为“true”时，才返回该参数。
layout_result	SmartDocumentRecognizerLayoutResult object	版面分析结果。当输入参数“layout”为“true”时，才返回该参数。
form_result	SmartDocumentRecognizerFormResult object	有线表单识别结果。当输入参数"form"为true时，才返回该参数。
formula_result	SmartDocumentRecognizerFormulaResult object	公式识别结果。

表 4-6 SmartDocumentRecognizerOcrResult

参数	参数类型	描述
direction	Float	图片朝向。
words_block_count	Integer	识别文字块数目。
words_block_list	Array of SmartDocumentRecognizerWordsBlockList objects	识别文字块列表，输出顺序从左到右，先上后下。

表 4-7 SmartDocumentRecognizerWordsBlockList

参数	参数类型	描述
words	String	文字块识别结果。
location	Array<Array<Integer>>	文字块的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的字段坐标仅用于参考，表示字段间的相对位置关系。
confidence	Float	文字块识别结果的置信度。

表 4-8 SmartDocumentRecognizerKvResult

参数	参数类型	描述
kv_block_count	Integer	模型识别到的键值对数量。
kv_block_list	Array of SmartDocumentRecognizerKVBlock objects	键值对识别结果列表。

表 4-9 SmartDocumentRecognizerKVBlock

参数	参数类型	描述
key	String	key-value对（键值对）中的key，例如“姓名：小明”中的“姓名”
value	String	key-value对（键值对）中的value，例如“姓名：小明”中的“小明”
words_block_count	Integer	该键值对中所包含的文本框数量。
words_block_list	Array of SmartDocumentRecognizerKVWordsBlock objects	文本框识别结果列表。

表 4-10 SmartDocumentRecognizerKVWordsBlock

参数	参数类型	描述
words	String	文字块识别结果。
location	Array<Array<Integer>>	文字块的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的字段坐标仅用于参考，表示字段间的相对位置关系。
type	String	类型。

表 4-11 SmartDocumentRecognizerTableResult

参数	参数类型	描述
table_count	Integer	模型识别到的表格数量。
table_list	Array of SmartDocumentRecognizerTableBlock objects	表格识别结果列表。

表 4-12 SmartDocumentRecognizerLayoutResult

参数	参数类型	描述
layout_block_count	Integer	模型识别到的文档版面区域数量。
layout_block_list	Array of SmartDocumentRecognizerLayoutBlock objects	文档版面区域识别结果列表。

表 4-13 SmartDocumentRecognizerLayoutBlock

参数	参数类型	描述
location	Array<Array<Integer>>	文字块的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的字段坐标仅用于参考，表示字段间的相对位置关系。

参数	参数类型	描述
type	String	文档区域类别，包含text、title、sub_title、image、image_caption、form、table、table_caption、header、footer、page_number、reference、formula、stamp、directory共15个类别。
text	String	文档区域文字内容。对于表格与图像，不返回其中的文字内容。
words_ids	Array of integers	文字识别结果索引列表，表示ocr_result的words_block_list中哪些文本框位于该文档区域内。
table_id	Integer	仅当type为"table"且入参table为True时返回该字段，表示当前逻辑表格区域对应table_result中哪一项识别结果。
form_id	Integer	仅当type为"form"且入参form为True时返回该字段，表示当前有线表单区域对应form_result中哪一项识别结果。

表 4-14 SmartDocumentRecognizerFormResult

参数	参数类型	描述
form_count	Integer	模型识别到的有线表单数量。
form_list	Array of SmartDocumentRecognizerTableBlock objects	有线表单识别结果列表。

表 4-15 SmartDocumentRecognizerTableBlock

参数	参数类型	描述
location	Array<Array<Integer>>	当前表格的位置信息，列表形式，分别表示文字块4个顶点的x, y坐标；坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的表格坐标仅用于参考，表示表格的相对位置关系。
words_block_count	Integer	表格中所包含的单元格数量。

参数	参数类型	描述
words_block_list	Array of SmartDocumentRecognizerTableWordsBlock objects	单元格识别结果列表。
excel	String	表格识别结果的base64编码，仅当return_excel为True时返回该字段。对返回的excel编码可用base64.b64decode解码并保存为.xlsx文件。

表 4-16 SmartDocumentRecognizerTableWordsBlock

参数	参数类型	描述
words	String	单元格内的文字识别结果。
rows	Array of integers	文字块占用的行信息，编号从0开始，列表形式，数据类型为Integer。
columns	Array of integers	文字块占用的列信息，编号从0开始，列表形式，数据类型为Integer。

表 4-17 SmartDocumentRecognizerFormulaResult

参数	参数类型	描述
formula_count	Integer	数学公式数量。
formula_list	Array of SmartDocumentRecognizerFormulaBlock objects	数学公式识别结果列表。

表 4-18 SmartDocumentRecognizerFormulaBlock

参数	参数类型	描述
formula	String	数学公式识别结果，以latex字符串表示。
location	Array<Array<Integer>>	数学公式位置信息，列表形式，分别表示4个顶点的x, y坐标；坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的数学公式坐标仅用于参考，表示数学公式的相对位置关系。

状态码： 400

表 4-19 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

请求示例

- 传入文档图片的base64编码进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/smart-document-recognizer

```
{
  "data": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
}
```
- 传入文档图片的url进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/smart-document-recognizer

```
{
  "url": "https://BucketName.obs.myhuaweicloud.com/ObjectName"
}
```

响应示例

状态码： 200

成功响应示例

```
{
  "result": [ {
    "formula_result": {
      "formula_count": 1,
      "formula_list": [ {
        "formula": "\\\int_{{L}} \\\left( 2 x y ^ { 3 } - y ^ { 2 } \\\cos x \\\right) \\\mathrm { d } x + \\\left( 1 - 2 y \\\sin x + 3 x ^ { 2 } y ^ { 2 } \\\right) \\\mathrm { d } y",
        "location": [ [ 171, 919 ], [ 950, 919 ], [ 950, 967 ], [ 171, 967 ] ]
      } ]
    }
  }, {
    "layout_result": {
      "layout_block_count": 19,
      "layout_block_list": [ {
        "location": [ [ 1165, 368 ], [ 2031, 368 ], [ 2031, 465 ], [ 1165, 465 ] ],
        "type": "title",
        "text": "黑龙江增值税专用发票",
        "words_ids": [ 0 ]
      }, {
        "location": [ [ 15, 19 ], [ 1078, 19 ], [ 1078, 637 ], [ 15, 637 ] ],
        "type": "form",
        "text": "xxxx",
        "words_ids": [ 2, 3, 4 ],
        "form_id": 0
      }, {
        "location": [ [ 18, 180 ], [ 1077, 180 ], [ 1077, 636 ], [ 18, 636 ] ],
        "type": "table",
        "text": "xxxx",
        "words_ids": [ 0, 1, 2 ],
        "table_id": 0
      } ]
    }
  } ]
}
```

```
    }
  }, {
    "form_result": {
      "form_count": 1,
      "form_list": [ {
        "location": [ [ 15, 19 ], [ 1074, 19 ], [ 1074, 636 ], [ 15, 636 ] ],
        "words_block_count": 24,
        "words_block_list": [ {
          "words": "xxx",
          "rows": [ 0 ],
          "columns": [ 0, 1, 2 ]
        }, {
          "words": "xxxx",
          "rows": [ 1 ],
          "columns": [ 0, 1, 2 ]
        } ],
        "excel": "UEsDBBQAAAAIAAAIQBhXUk6TwEAAI8EAAATAAAAW0NvbnRlbnRfVHlwZX..."
      } ]
    }
  }, {
    "table_result": {
      "table_count": 1,
      "table_list": [ {
        "words_block_count": 24,
        "words_block_list": [ {
          "words": "货物或应税劳务、服务名称",
          "rows": [ 0 ],
          "columns": [ 0 ]
        }, {
          "words": "规格型号",
          "rows": [ 0 ],
          "columns": [ 1 ]
        } ],
        "excel": "xxxx",
        "location": [ [ 275, 967 ], [ 2919, 967 ], [ 2919, 1177 ], [ 275, 1177 ] ]
      } ]
    }
  }, {
    "kv_result": {
      "kv_block_count": 25,
      "kv_block_list": [ {
        "key": "开票日期",
        "value": "2017年08月31日",
        "words_block_count": 2,
        "words_block_list": [ {
          "words": "开票日期",
          "location": [ [ 2241, 589 ], [ 2480, 592 ], [ 2480, 646 ], [ 2241, 643 ] ],
          "type": "key"
        }, {
          "words": "2017年08月31日",
          "location": [ [ 2479, 591 ], [ 2850, 595 ], [ 2850, 649 ], [ 2479, 645 ] ],
          "type": "value"
        } ]
      } ]
    }
  }, {
    "ocr_result": {
      "direction": 0.4767,
      "words_block_count": 67,
      "words_block_list": [ {
        "words": "黑龙江增值税专用发票",
        "location": [ [ 430, 100 ], [ 874, 99 ], [ 874, 139 ], [ 430, 141 ] ],
        "confidence": 0.9552
      } ]
    }
  }
}
```

状态码: 400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入文档图片的base64编码进行文字识别

```
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 RecognizeSmartDocumentRecognizerSolution {

    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();
        RecognizeSmartDocumentRecognizerRequest request = new
        RecognizeSmartDocumentRecognizerRequest();
        SmartDocumentRecognizerRequestBody body = new SmartDocumentRecognizerRequestBody();
        body.withData("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeSmartDocumentRecognizerResponse response =
            client.recognizeSmartDocumentRecognizer(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());  
    }  
}
```

- 传入文档图片的url进行文字识别

```
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 RecognizeSmartDocumentRecognizerSolution {
```

```
    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();  
        RecognizeSmartDocumentRecognizerRequest request = new  
        RecognizeSmartDocumentRecognizerRequest();  
        SmartDocumentRecognizerRequestBody body = new SmartDocumentRecognizerRequestBody();  
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");  
        request.withBody(body);  
        try {  
            RecognizeSmartDocumentRecognizerResponse response =  
            client.recognizeSmartDocumentRecognizer(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

- 传入文档图片的base64编码进行文字识别

```
# 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 = RecognizeSmartDocumentRecognizerRequest()
        request.body = SmartDocumentRecognizerRequestBody(
            data="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_smart_document_recognizer(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 传入文档图片的url进行文字识别

```
# 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 = RecognizeSmartDocumentRecognizerRequest()
        request.body = SmartDocumentRecognizerRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_smart_document_recognizer(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

- 传入文档图片的base64编码进行文字识别

```
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.RecognizeSmartDocumentRecognizerRequest{
        dataSmartDocumentRecognizerRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
        request.Body = &model.SmartDocumentRecognizerRequestBody{
            Data: &dataSmartDocumentRecognizerRequestBody,
        }
    }
    response, err := client.RecognizeSmartDocumentRecognizer(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 传入文档图片的url进行文字识别

```
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.RecognizeSmartDocumentRecognizerRequest{}
urlSmartDocumentRecognizerRequestBody:= "https://BucketName.obs.myhuaweicloud.com/
ObjectName"
request.Body = &model.SmartDocumentRecognizerRequestBody{
    Url: &urlSmartDocumentRecognizerRequestBody,
}
response, err := client.RecognizeSmartDocumentRecognizer(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.2 通用证件识别

 说明

OCR通用证件识别将于2025.11.04正式下架，为避免影响您目前的业务，请您及时进行相关迁移工作。

功能介绍

识别身份证件图像，并将识别的结构化结果返回给用户。支持多个国家/地区的身份证、驾驶证和护照，具体国家/地区和证件列表详见[表4-20](#)。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

表 4-20 国家/地区和证件列表

国家/地区	英文名称	国家/地区代码 country_region	支持证件类型 id_type
越南	Vietnam	VNM	PP、DL、ID
印度	India	IND	PP
菲律宾	Philippines	PHL	PP、DL、ID（仅支持UMID）
阿尔巴尼亚	Albania	ALB	PP、DL、ID
巴西	Brazil	BRA	PP
印度尼西亚	Indonesia	IDN	PP
马来西亚	Malaysia	MYS	PP
尼日利亚	Nigeria	NGA	PP
巴基斯坦	Pakistan	PAK	PP
俄罗斯	Russia	RUS	PP（仅支持国际标准版本）
中国-台湾	Taiwan (China)	TWN	PP
乌克兰	Ukraine	UKR	PP
泰国	Thailand	THA	ID、PP
智利	Chile	CHL	ID、PP
中国-香港	Hong Kong (China)	HKG	ID
● PP: passport，国际护照。 ● DL: driving license，驾驶证。 ● ID: identification card，各国/地区颁发的身份证类型证件，比如身份证、选民卡、社保卡等。			

约束与限制

- 只支持JPEG、JPG、PNG、BMP、TIFF格式图片。
- 图片最小边不小于100px，最长边不超过8192px。单个图片对应的Base64编码不超过10MB。

调用方法

请参见[如何调用API](#)。

前提条件

在使用通用证件识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-21 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-22 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

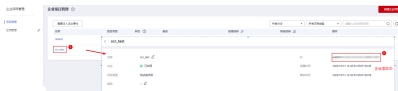
参数	是否必选	参数类型	描述
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 - 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 - 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 - 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-23 请求 Body 参数

参数	是否必选	参数类型	描述
image	否	String	该参数与url二选一。图像数据，base64编码，要求base64编码后大小不超过10MB。图片最小边不小于100px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF。

参数	是否必选	参数类型	描述
url	否	String	该参数与image二选一。图片的url路径，目前支持： Image URL. Currently, the following URLs are supported: • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。
country_region	否	Array of strings	证件签发国家或地区代码，命名遵循ISO-3166 3位代码。可选值。支持填写1个或多个国家/地区。指定参数后，服务只识别指定国家/地区的卡证，如留空，则识别所有支持的卡证。建议国家/地区固定或有限范围的情况下填写。支持国家/地区列表见表4-20。 说明 识别越南语时，该参数为必填。
id_type	否	Array of strings	证件类型。可选值。支持填写1种或多种证件。指定参数后，服务只识别指定类型的卡证，如留空，默认识别所有类型卡证，建议已知证件类型的情况下填写。支持证件类型如下： • PP: passport，国际护照。 • DL: driving license，驾驶证。 • ID: identification card，各国/地区颁发的身份证类型证件，比如身份证、选民卡、社保卡等。
return_portrait_image	否	Boolean	控制是否返回portrait_image（证件中的人像图片），true代表需要返回，false代表不需要。

响应参数

状态码： 200

表 4-24 响应 Body 参数

参数	参数类型	描述
result	IdDocumentItem object	识别结果。 调用失败时此字段为空。

表 4-25 IdDocumentItem

参数	参数类型	描述
country_region	String	证件签发国家或地区代码，命名遵循ISO-3166 3位代码。当前支持国家/地区列表见 表4-20 。
id_type	String	证件类型，可选值如下： - PP: passport，国际护照。 - DL: driving license，驾驶证。 - ID: identification card，各国/地区颁发的身份证类型证件，比如身份证、选民卡、社保卡等。
side	String	证件正面或反面,可选值： - front: 正面，一般是包含人像的那面。 - back: 背面 对于只有一面的卡证，返回front。
first_name	String	名。
last_name	String	姓氏。
sex	String	性别，可选值： M:男性 F:女性 X:中性。
nationality	String	持有人国籍。
birth_date	String	生日，格式为yyyy-mm-dd。
issue_date	String	签发日期，yyyy-mm-dd。
expiry_date	String	有效日期，yyyy-mm-dd。
document_number	String	证件号码。
address	String	持有人通讯地址。
issuing_authority	String	签发机关。
portrait_image	String	可选返回，证件头像图像base64编码。

参数	参数类型	描述
confidence	Object	字段置信度，为0~1之间的小数，值越大，表明识别结果越可靠。

状态码： 400

表 4-26 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，通用证件识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com”或“ocr.ap-southeast-1.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/id-document”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)
- 如何获取Token请参见[构造请求](#)。
- 请求示例（方式一：使用图片的Base64编码）
POST https://{endpoint}/v2/{project_id}/ocr/id-document

```
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "country_region": "ALB",
  "id_type": "PP"
}
```
- 请求示例（方式二：使用图片URL）
POST https://{endpoint}/v2/{project_id}/ocr/id-document

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

响应示例

状态码： 200

成功响应示例

```
{
  "result": {
    "country_region": "ALB",
    "id_type": "PP",
```

```
"side" : "front"
}
}
```

状态码： 400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入证件图片的base64编码进行文字识别

```
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());
}
}
```

- 传入证件图片的url进行文字识别

```
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.xxx.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

- 传入证件图片的base64编码进行文字识别

```
# 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)
```

- 传入证件图片的url进行文字识别

```
# 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.xxx.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

- 传入证件图片的base64编码进行文字识别

```
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)
}
}
```

- 传入证件图片的url进行文字识别

```
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.xxx.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)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.3 智能证件分类

功能介绍

支持9类证件的分类和告警检测，以JSON格式返回结果。支持的证件类型有秘鲁身份证、柬文身份证、香港身份证、澳门身份证、缅文身份证、缅文驾驶证、泰文身份证、护照和中华人民共和国居民身份证。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/ocr/auto-id-doc-classification

表 4-27 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-28 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“ 企业项目管理 ”页面，单击企业项目名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。 企业项目创建步骤请参见用户指南。 说明 说明： 创建企业项目后，在传参时，有以下三类场景。 ● 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 ● 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 ● 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-29 请求 Body 参数

参数	是否必选	参数类型	描述
data	否	String	该参数与url二选一。 图片的Base64编码，单个图片其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片尺寸不小于15px，最长边不超过8192px，支持JPEG/JPG/PNG/BMP/TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。
url	否	String	与data二选一。 单个图片其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议文件大小不超过7MB。 图片尺寸不小于15px，最长边不超过8192px，支持JPEG/JPG/PNG/BMP/TIFF格式。 图片的URL路径，目前支持： ● 公网http/https url ● OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 ● 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 ● 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 ● url中不能存在中文字符，若存在，中文需要进行utf8编码

参数	是否必选	参数类型	描述
alarm	否	Boolean	证件告警检测功能的开关，默认 false，取值范围如下： • true：开启证件图像告警功能。 • false：关闭证件图像告警功能。

响应参数

状态码：200

表 4-30 响应 Body 参数

参数	参数类型	描述
result	AutoldDocClassificationResult object	调用成功时表示调用结果。 调用失败时无此字段。

表 4-31 AutoldDocClassificationResult

参数	参数类型	描述
type	String	证件的类型，支持的证件类型如下： • peru_id_card：秘鲁身份证 • cambodian_id_card：柬文身份证 • hongkong_id_card：香港身份证 • macao_id_card：澳门身份证 • myanmar_driver_license：缅文驾驶证 • myanmar_id_card：缅文身份证 • passport：护照 • thailand_id_card：泰文身份证 • id_card：中华人民共和国居民身份证
location	Array<Array<Integer>>	证件的位置。
confidence	Float	证件位置的置信度。

参数	参数类型	描述
alarm_result	Object	图像告警检测结果。 仅在输入参数alarm为true时，返回该字段，包含如下字段，各字段解释请参见表4-32。 detect_blur_result、 detect_glare_result、 detect_blocking_within_border_result、 detect_insufficient_lighting_result、 detect_copy_result、 detect_border_integrity_result、 detect_reproduce_result
alarm_confidence	Object	图像告警检测结果分数。 仅在输入参数alarm为true时，返回该字段，包含字段如下，各字段解释请参见表4-33。 ：blur_score、glare_score、 blocking_within_border_score、 insufficient_lighting_score、 copy_score、border_integrity_score、 reproduce_score，以上字段取值范围为[0, 99]。

表 4-32 AutoldDocClassificationAlarmResult

参数	参数类型	描述
detect_blur_result	Boolean	证件图像模糊告警结果。 ● true：表示证件图片较模糊。 ● false：表示证件清晰。
detect_glare_result	Boolean	证件图像反光告警结果。 ● true：表示证件图片较模糊。 ● false：表示证件清晰。
detect_blocking_within_border_result	Boolean	证件图像框内遮挡告警结果。 ● true：表示证件图片较模糊。 ● false：表示证件清晰。
detect_insufficient_lighting_result	Boolean	证件图像过暗告警结果。 ● true：表示证件图片较模糊。 ● false：表示证件清晰。

参数	参数类型	描述
detect_copy_result	Boolean	证件图像是否黑白复印件告警结果。 - true：表示证件是复印件。 - false：表示证件是原件。
detect_border_integrity_result	Boolean	证件图像边框完整性告警结果。 - true：表示边框不完整。 - false：表示边框完整。
detect_reproduce_result	Boolean	证件图像是否翻拍告警结果。 - true：表示证件图片经过翻拍。 - false：表示证件图片未经过翻拍。

表 4-33 AutoldDocClassificationAlarmConfidence

参数	参数类型	描述
blur_score	Integer	证件图像模糊告警分数，分数越高，证件图像模糊的可能性越高。
glare_score	Integer	证件图像反光告警分数，分数越高，证件图像反光的可能性越高。
blocking_within_border_score	Integer	证件图像框内遮挡告警分数，分数越高，证件图像框内遮挡的可能性越高。
insufficient_lighting_score	Integer	证件图像过暗告警分数，分数越高，证件图像过暗的可能性越高。
copy_score	Integer	证件图像复印件告警分数，分数越高，证件图像是复印件的可能性越高。
border_integrity_score	Integer	证件图像边框完整性告警分数，分数越高，证件图像边框不完整的可能性越高。
reproduce_score	Integer	证件图像翻拍告警分数，分数越高，证件图像经过翻拍的可能性越高。

状态码：400

表 4-34 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见错误码。 调用成功时无此字段。

参数	参数类型	描述
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

请求示例

- 传入智能证件分类图片的base64编码进行文字识别

```
POST https://{endpoint}/v2/{project_id}/ocr/auto-id-doc-classification

{
  "data": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "alarm": true
}
```

- 传入智能证件分类图片的url进行文字识别

```
POST https://{endpoint}/v2/{project_id}/ocr/auto-id-doc-classification

{
  "url": "https://BucketName.obs.myhuaweicloud.com/ObjectName",
  "alarm": true
}
```

响应示例

状态码：200

成功响应示例

```
{
  "result": {
    "type": "id_card",
    "location": [ [ 275, 125 ], [ 860, 165 ], [ 916, 634 ], [ 255, 633 ] ],
    "confidence": 0.979,
    "alarm_result": {
      "detect_blur_result": true,
      "detect_glare_result": false,
      "detect_blocking_within_border_result": false,
      "detect_insufficient_lighting_result": false,
      "detect_copy_result": false,
      "detect_border_integrity_result": false,
      "detect_reproduce_result": false
    },
    "alarm_confidence": {
      "blur_score": 98,
      "glare_score": 0,
      "blocking_within_border_score": 0,
      "insufficient_lighting_score": 0,
      "copy_score": 0,
      "border_integrity_score": 0,
      "reproduce_score": 0
    }
  }
}
```

状态码：400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

Java

- 传入智能证件分类图片的base64编码进行文字识别

```
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 RecognizeAutoldDocClassificationSolution {

    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");
        String projectId = "{project_id}";

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

        OcrClient client = OcrClient.newBuilder()
            .withCredential(auth)
            .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
            .build();

        RecognizeAutoldDocClassificationRequest request = new
        RecognizeAutoldDocClassificationRequest();
        AutoldDocClassificationRequestBody body = new AutoldDocClassificationRequestBody();
        body.withAlarm(true);
        body.withData("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
        request.withBody(body);
        try {
            RecognizeAutoldDocClassificationResponse response =
            client.recognizeAutoldDocClassification(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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 传入智能证件分类图片的url进行文字识别

```
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 RecognizeAutoldDocClassificationSolution {

    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");
        String projectId = "{project_id}";

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

        OcrClient client = OcrClient.newBuilder()
            .withCredential(auth)
            .withRegion(OcrRegion.valueOf("<YOUR REGION>"))
            .build();
        RecognizeAutoldDocClassificationRequest request = new
        RecognizeAutoldDocClassificationRequest();
        AutoldDocClassificationRequestBody body = new AutoldDocClassificationRequestBody();
        body.withAlarm(true);
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeAutoldDocClassificationResponse response =
            client.recognizeAutoldDocClassification(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

- 传入智能证件分类图片的base64编码进行文字识别

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = RecognizeAutoldDocClassificationRequest()
    request.body = AutoldDocClassificationRequestBody(
        alarm=True,
        data="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    )
    response = client.recognize_auto_id_doc_classification(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 传入智能证件分类图片的url进行文字识别

coding: utf-8

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = RecognizeAutoldDocClassificationRequest()
        request.body = AutoldDocClassificationRequestBody(
            alarm=True,
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_auto_id_doc_classification(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

- 传入智能证件分类图片的base64编码进行文字识别

```
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")
    projectId := "{project_id}"

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

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

    request := &model.RecognizeAutoldDocClassificationRequest{
        alarmAutoldDocClassificationRequestBody:= true
        dataAutoldDocClassificationRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
        request.Body = &model.AutoldDocClassificationRequestBody{
            Alarm: &alarmAutoldDocClassificationRequestBody,
            Data: &dataAutoldDocClassificationRequestBody,
        }
    }
    response, err := client.RecognizeAutoldDocClassification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 传入智能证件分类图片的url进行文字识别

```
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")
projectId := "{project_id}"

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

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

request := &model.RecognizeAutoldDocClassificationRequest{
    alarmAutoldDocClassificationRequestBody:= true
    urlAutoldDocClassificationRequestBody:= "https://BucketName.obs.myhuaweicloud.com/
ObjectName"
    request.Body = &model.AutoldDocClassificationRequestBody{
        Alarm: &alarmAutoldDocClassificationRequestBody,
        Url: &urlAutoldDocClassificationRequestBody,
    }
    response, err := client.RecognizeAutoldDocClassification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.4 通用表格识别

功能介绍

识别表格图片中的文字内容，并将识别的结果以JSON格式返回给用户。返回结果将图片区域（ words_region ）划分为两类：纯文本区（ text ）和表格区（ table ），并返回表格结构（ row, column ）和文本信息。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

约束与限制

- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式图片。
- 图像各边的像素大小在15px到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中识别区域有效占比超过80%，保证整张表格及其边缘包含在图像内。
- 支持图像任意角度的水平旋转。
- 目前不支持复杂背景（如户外自然场景、防伪水印等）和表格线扭曲图像的文字识别。
- 支持中英文以及部分繁体字。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts 4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-35 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-36 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 • 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 • 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 • 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-37 请求 Body 参数

参数	是否必选	类型	说明
image	否	String	该参数与url二选一。 单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片最短边不小于15px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAQ...，带有多余前缀会产生The image format is not supported报错。
url	否	String	该参数与image二选一。 url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片的url路径目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 • url中不能存在中文字符，若存在，中文需要进行utf8编码。
return_text_location	否	Boolean	返回文本块坐标及单元格坐标信息，可选值如下所示。 • true：返回文本块和单元格坐标 • false：不返回 如果未传入该参数时默认为false，即不返回。
return_character_location	否	Boolean	返回单字符的坐标信息，可选值包括： • true：返回单字符的坐标 • false：不返回 未传入该参数时默认为false，即不返回。如果此参数为true时，return_text_location必须为true

参数	是否必选	类型	说明
return_confidence	否	Boolean	是否返回置信度的开关，可选值如下所示。 - true: 返回置信度 - false: 不返回置信度 如果未传入该参数，系统默认为“false”，即不返回置信度。
return_excel	否	Boolean	是否返回表格转换Microsoft Excel的Base64编码字段。可选值包括： - true: 返回'excel'字段，表示xlsx格式的表格识别结果的Base64编码 - false: 不返回。默认为false 对返回的Excel编码，可用Python函数base64.b64decode解码后保存为.xlsx文件。
return_rectification_matrix	否	Boolean	可选值包括： - true: 返回透视变换矩阵 - false: 不返回 未传入该参数时默认为false，即不返回透视变换矩阵。
with_borders	否	Boolean	可选值包括： - true: 输入图像仅包含有线表格，仅进行有线表格识别。 - false: 输入图像可能包含无线表格，同时进行有线表格与无线表格识别。 未传入该参数时默认为false，即同时进行有线表格与无线表格识别。当确认输入仅包含有线表格时，该参数设为true可达到更优识别效果。

响应参数

说明

根据识别的结果，可能有不同的HTTP响应状态码（status code）。例如，200表示API调用成功，400表示调用失败，详细的状态码和响应参数说明如下。

状态码： 200

表 4-38 响应 Body 参数

参数	参数类型	描述
result	GeneralTable Result object	调用成功时表示调用结果。 调用失败时无此字段。

表 4-39 GeneralTableResult

参数	参数类型	描述
words_region_count	Integer	文字区域数目。
words_region_list	Array of WordsRegionList objects	文字区域识别结果列表，输出顺序从左到右，先上后下。
excel	String	表格图像转换为excel的base64编码，图像中的文字和表格按位置写入excel。对返回的excel编码可用base64.b64decode解码并保存为.xlsx文件。

表 4-40 WordsRegionList

参数	参数类型	描述
type	String	文字识别区域类型。 • text：文本识别区域; • table：表格识别区域。
words_block_count	Integer	子区域识别文字块数目。
words_block_list	Array of GeneralTableWordsBlockList objects	子区域识别文字块列表，输出顺序从左到右，先上后下。

表 4-41 GeneralTableWordsBlockList

参数	参数类型	描述
words	String	文字块识别结果。
confidence	Float	字段的平均置信度，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于对应字段的准确率。
location	Array<Array<Integer>>	文字块位置信息，列表形式，分别表示文字块4个顶点的x, y坐标;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。
words_list	Array of WordsListItem objects	单元格内文字段列表。输出顺序从左到右，从上到下。仅当入参"return_text_location"为true时存在。

参数	参数类型	描述
rows	Array of integers	文字块占用的行信息，编号从0开始，列表形式，数据类型为Integer。仅在表格区域内有效，即type字段为"table"时该字段有效。
columns	Array of integers	文字块占用的列信息，编号从0开始，列表形式，数据类型为Integer。仅在表格区域内有效，即type字段为"table"时该字段有效。
cell_location	Array<Array<Integer>>	单元格位置信息，列表形式，分别表示单元格4个顶点的x, y坐标;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。

表 4-42 WordsListItem

参数	参数类型	描述
words	String	文字块识别结果。
confidence	Float	字段的平均置信度，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于对应字段的准确率。
location	Array<Array<Integer>>	文字块位置信息，列表形式，分别表示文字块4个顶点的x, y坐标;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。
char_list	Array of CharListItem objects	单元格内文字段列表。输出顺序从左到右，从上到下。仅当入参"return_text_location"和"return_char_location"同时为true时存在。

表 4-43 CharListItem

参数	参数类型	描述
char	String	单字符识别结果。
char_confidence	Float	单字符的置信度，置信度越大，表示本次识别的文字的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于对应字段的准确率。
char_location	Array<Array<Integer>>	单字符的位置信息，列表形式，分别表示文字块4个顶点的x, y坐标;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。

状态码： 400

表 4-44 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，通用表格识别服务部署在“亚太-新加坡”区域的“endpoint”为“ocr.ap-southeast-3.myhuaweicloud.com”或“ocr.ap-southeast-3.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-3.myhuaweicloud.com/v2/{project_id}/ocr/general-table”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token请参见[认证鉴权](#)。
- 传入表格图片的base64编码文字识别，并不返回置信度信息
POST https://{endpoint}/v2/{project_id}/ocr/general-table
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:
{
 "image": "/9j/4AAQSkZJRgABAQEAASABIAAD/4RFZRXhpZgAATU0AKgAAAAg...",
 "return_confidence": false
}
- 传入表格图片的url进行文字识别，并不返回置信度信息
POST https://{endpoint}/v2/{project_id}/ocr/general-table
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:
{
 "url": "https://BucketName.obs.xxx.com/ObjectName",
 "return_confidence": false
}

响应示例

状态码： 200

成功响应示例

```
{  
  "result": {  
    "words_region_count": 2,  
    "words_region_list": [ {  
      "type": "text",  
      "words_block_count": 1,  
      "words_block_list": [ {  
        "words": "文字区域识别文字块1",  
        "confidence": 0.9991  
      } ]  
    } ]  
  }, {
```

```
{
  "type": "table",
  "words_block_count": 2,
  "words_block_list": [ {
    "words": "表格区域识别文字块1",
    "confidence": 0.9942,
    "rows": [ 0 ],
    "columns": [ 0 ]
  }, {
    "words": "表格区域识别文字块2",
    "confidence": 0.914,
    "rows": [ 0 ],
    "columns": [ 1, 2 ]
  } ]
}
```

状态码： 400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入表格图片的base64编码进行文字识别，并不返回置信度信息

```
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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 传入表格图片的url进行文字识别，并不返回置信度信息

```
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();
        }
    }
}
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- 传入表格图片的base64编码进行文字识别，并不返回置信度信息

```
# 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)
```

- 传入表格图片的url进行文字识别，并不返回置信度信息

```
# 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

- 传入表格图片的base64编码进行文字识别，并不返回置信度信息

```
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)
    }
}
```

- 传入表格图片的url进行文字识别，并不返回置信度信息

```
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)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.5 通用文字识别

功能介绍

识别图片上的文字信息，以JSON格式返回识别的文字和坐标。支持扫描文件、电子文档、书籍、票据和表单等多种场景的文字识别。

支持中英文以及部分繁体字。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

约束与限制

- 只支持识别PNG、JPG、JPEG、BMP、GIF、TIFF、WEBP、PCX、ICO、PSD、PDF格式图片。
- 图像各边的像素大小在15px到30000px之间，图像高*宽的总像素数不能大于1.6亿。单个图片、PDF文件其对应的Base64编码不超过10MB。
- 图像中识别区域有效占比超过80%，保证所有文字及其边缘包含在图像内。
- 支持图像任意角度的水平旋转。
- 支持自动过滤浅色文字水印。
- 目前不支持复杂背景（如户外自然场景等）和文字扭曲图像的文字识别。
- 支持中英文以及部分繁体字、马来语、乌克兰语、印地语、俄语、越南语、印尼语、泰语、阿拉伯语、德语、拉丁语、法语、意大利语、西班牙语、葡萄牙语、罗马尼亚语、波兰语、阿姆哈拉语、日语、韩语、土耳其语、挪威语、丹麦语、瑞典语、柬埔寨语、希伯来语识别。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-45 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-46 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

参数	是否必选	参数类型	描述
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 - 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 - 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 - 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-47 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	该参数与url二选一。 Base64编码，要求单个图片、PDF文件其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议文件大小不超过7MB。 图片最短边不小于15px，最长边不超过30000px。支持JPEG、JPG、PNG、BMP、GIF、TIFF、WEBP、PCX、ICO、PSD、PDF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAQ...，带有多余前缀会产生The image format is not supported报错。

参数	是否必选	参数类型	说明
url	否	String	该参数与image二选一。 单个图片、PDF文件其对应的Base64编码不超过10MB。文件在Base64编码后会大于文件原本大小，请注意做好边界判断，建议文件大小不超过7MB。 图片的url路径目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 • url中不能存在中文字符，若存在，中文需要进行utf8编码。
detect_direction	否	Boolean	是否校正图片的倾斜角度，可选值如下。 • true：校正图片的倾斜角度 • false：不校正图片的倾斜角度 支持任意角度的校正，未传入该参数时默认为“false”。 待识别图片如果存在倾斜，建议将此参数设置为“true”。
quick_mode	否	Boolean	快速模式开关，针对单行文字图片（要求图片只包含一行文字，且文字区域占比超过50%），打开时可以更快返回识别内容。可选值如下所示。 • true：打开快速模式 • false：关闭快速模式 未传入该参数时默认为false，即关闭快速模式。
character_mode	否	Boolean	单字符模式开关。可选值包括： • true：打开单字符模式 • false：关闭单字符模式 未传入该参数时默认为false，即不返回单个文本行的单字符信息。

参数	是否必选	参数类型	说明
language	否	String	语种选择，未传入该参数时默认为中英文识别模式。 • auto：自动语种分类 • ms：马来语 • uk：乌克兰语 • hi：印地语 • ru：俄语 • vi：越南语 • id：印尼语 • th：泰语 • zh：中英文 • ar：阿拉伯语 • de：德语 • la：拉丁语 • fr：法语 • it：意大利语 • es：西班牙语 • pt：葡萄牙语 • ro：罗马尼亚语 • pl：波兰语 • am：阿姆哈拉语 • ja：日语 • ko：韩语 • tr：土耳其语 • no：挪威语 • da：丹麦语 • sv：瑞典语 • km：柬埔寨语 • he：希伯来语
single_orientation_mode	否	Boolean	单朝向模式开关。可选值包括： • true：打开单朝向模式 • false：关闭单朝向模式 未传入该参数时默认为false，即默认图片中的字段为多朝向。
pdf_page_number	否	Integer	指定PDF页码识别。传入该参数时，则识别指定页码的内容。如果不传该参数，则默认识别第1页。

参数	是否必选	参数类型	说明
return_markdown_result	否	Boolean	返回文字块拼接结果开关。可选值包括： • true：打开返回文字块拼接结果开关； • false：关闭返回文字块拼接结果开关。 未传入该参数时默认为false，既默认关闭返回文字块拼接结果开关。

响应参数

说明

根据识别的结果，可能有不同的HTTP响应状态码（status code）。例如，200表示API调用成功，400表示调用失败，详细的状态码和响应参数说明如下。

状态码： 200

表 4-48 响应 Body 参数

参数	参数类型	描述
result	GeneralTextResult object	识别结果。 调用失败时不返回此字段。

表 4-49 GeneralTextResult

参数	参数类型	描述
direction	Float	图片朝向。 • 当“detect_direction”为“true”时，该字段有效。返回图片逆时针旋转角度，值区间为0~359。 • 当“detect_direction”为“false”时，该字段值为 -1。
words_block_count	Integer	检测到的文字块数目。
words_block_list	Array of GeneralTextWordsBlockList objects	识别文字块列表。输出顺序从左到右，先上后下。

表 4-50 GeneralTextWordsBlockList

参数	参数类型	描述
words	String	文字块识别结果。
location	Array<Array<Integer>>	文字块的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 说明 输入数据格式是PDF时，返回的字段坐标仅用于参考，表示字段间的相对位置关系。
confidence	Float	文字块识别结果的置信度。
char_list	Array of GeneralTextCharList objects	文字块对应的单字符识别列表，输出顺序从左到右，先上后下。

表 4-51 GeneralTextCharList

参数	参数类型	描述
char	String	单字符识别结果。
char_location	Array<Array<Integer>>	单字符的区域位置信息，列表形式，包含字符区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。
char_confidence	Float	单字符识别结果的置信度。
markdown_result	String	所有文字块拼接的识别结果，同一行的文字块使用“\t”拼接，不同行的文字块使用“\n”拼接。 当return_markdown_result为true时，返回该字段值，否则，不返回该字段。

状态码： 400

表 4-52 响应 Body 参数

参数	参数类型	说明
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时返回的错误信息。 调用成功时不返回此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，通用文字识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com”或“ocr.ap-southeast-1.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/general-text”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token请参见[认证鉴权](#)。
- 传入图片的base64编码进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式
POST https://{endpoint}/v2/{project_id}/ocr/general-text
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:

```
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRxpZgAATU0AKgAAAA...",
  "detect_direction": false,
  "quick_mode": false
}
```
- 传入图片的url进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式
POST https://{endpoint}/v2/{project_id}/ocr/general-text
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:

```
{
  "url": "https://BucketName.obs.xxx.com/ObjectName",
  "detect_direction": false,
  "quick_mode": false
}
```

响应示例

状态码：200

成功响应示例

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

状态码：400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入图片的base64编码进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式

```
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/4AAQSkZJRgABAQgEASABIAAD/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());
    }
}
```

- 传入图片的url进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式

```
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

- 传入图片的base64编码进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式

```
# 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)
```

- 传入图片的url进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式
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

- 传入图片的base64编码进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式

```
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)
    }
}
```

- 传入图片的url进行文字识别，识别过程不校验图片倾斜角度，并关闭快速模式

```
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)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.6 网络图片识别

功能介绍

识别网络图片中的文字内容，并以JSON格式返回识别的结构化结果。

该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

约束与限制

- 支持中英文及部分中文繁体字。
- 只支持识别JPG、JPEG、PNG、BMP、TIFF、TGA、WEBP、ICO、PCX、GIF格式图片。
- 支持常见网络图片：手机截图、电脑截图、电商产品图及广告设计图等互联网图片。
- 图像各边的像素大小在15px到30000px之间。单个图片对应的Base64编码不超过10MB。
- 图像中有效文字图片占比超过60%，避免有效文字图片占比过小。
- 支持图像中有效文字图片的任意角度的水平旋转（需开启方向检测）。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-53 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

表 4-55 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	该参数与url二选一。 单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片最短边不小于15px，最长边不超过30000px，支持JPG、JPEG、PNG、BMP、TIFF、TGA、WEBP、ICO、PCX、GIF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。
url	否	String	该参数与image二选一。url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片的url路径目前支持： - 公网http/https url - OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 - 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 - 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 - url中不能存在中文字符，若存在，中文需要进行utf8编码。

参数	是否必选	参数类型	说明
detect_direction	否	Boolean	是否校正图片的倾斜角度，可选值如下。 - true：校正图片的倾斜角度 - false：不校正图片的倾斜角度 支持任意角度的校正，未传入该参数时默认为“false”。待识别图片如果存在倾斜，建议将此参数设置为“true”。
extract_type	否	Array of strings	结构化数据提取参数列表，目前只支持图像宽高，其入参值为“image_size”。若不填写该参数或删除该参数，默认不提取该参数值。
detect_font	否	Boolean	为Boolean类型，若不传该字段，默认不检测切片字体，为True时，将检测切片的字体类型，并返回最相似的5种字体名称。
detect_text_direction	否	Boolean	为Boolean类型，若不传该字段，默认为True，即检测每个字段的文字方向。为False时，则不检测文字方向。若图片中所有文字方向均是水平朝上时，建议将该值设为False，即不检测文字方向。

响应参数

说明

根据识别的结果，可能有不同的HTTP响应状态码（status code）。例如，200表示API调用成功，400表示调用失败，详细的状态码和响应参数说明如下。

状态码： 200

表 4-56 响应 Body 参数

参数	参数类型	描述
result	WebImageResult object	调用成功时表示调用结果。 调用失败时无此字段。

表 4-57 WebImageResult

参数	参数类型	描述
words_block_count	Integer	代表检测识别出来的文字块数目。

参数	参数类型	描述
words_block_list	Array of WebImageWordsBlockList objects	识别文字块列表，输出顺序从左到右，从上到下。
extracted_data	WebImageExtractedData object	提取出的结构化JSON结果，该字典内的key值与入参列表extract_type的值一致，目前仅支持联系人、图像高宽信息提取，亦即key值为"contact_info"的字段、"image_size"字段。 若入参extract_type为空列表或该字段缺失时，不进行提取，此字段为空。

表 4-58 WebImageWordsBlockList

参数	参数类型	描述
words	String	文字块识别结果。
confidence	Float	相关字段的置信度信息，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于对应字段的准确率。
location	Array<Array<Integer>>	文字块的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。
font_list	Array of strings	文字块所属字体类型，列表形式，表示与文字块的文字最接近的字体类型。
font_scores	Array of numbers	文字块所属字体类型的概率，列表形式，与font_list一一对应，表示文字块的文字属于某种字体类型的概率。

表 4-59 WebImageExtractedData

参数	参数类型	描述
contact_info	WebImageContactInfo object	该字段表示提取的联系人信息，包括：姓名、联系电话、省市区以及详细地址。若入参extract_type列表中无该字段，则此字段不存在。
image_size	WebImageImageSize object	该字段表示返回图片宽高信息。如入参extract_type列表中无该字段，则此字段不存在。

表 4-60 WeblImageContactInfo

参数	参数类型	描述
name	String	传入contact_info时的返回，为姓名。
phone	String	传入contact_info时的返回，联系电话。
province	String	传入contact_info时的返回，省。
city	String	传入contact_info时的返回，市。
district	String	传入contact_info时的返回，县区。
detail_address	String	传入contact_info时的返回，详细地址（不含省市区）。

表 4-61 WeblImageImageSize

参数	参数类型	描述
height	Integer	传入image_size时的返回，为图像高度。
width	Integer	传入image_size时的返回，为图像宽度。

状态码： 400

表 4-62 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，网络图片识别服务部署在“亚太-曼谷”区域的“endpoint”为“ocr.ap-southeast-2.myhuaweicloud.com”或“ocr.ap-southeast-2.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/ocr/web-image”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token请参见[认证鉴权](#)。
- 传入网络图片的base64编码进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/web-image
Request Header:
Content-Type: application/json

```
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwggguVBgkqhkiG...
```

Request Body:

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

- 传入网络图片的url进行文字识别

POST https://{endpoint}/v2/{project_id}/ocr/web-image

Request Header:

Content-Type: application/json

X-Auth-Token:

```
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwggguVBgkqhkiG...
```

Request Body:

```
{
  "url": "https://BucketName.obs.xxx.com/ObjectName"
}
```

响应示例

状态码：200

成功响应示例

```
{
  "result": {
    "words_block_count": 3,
    "words_block_list": [
      {
        "words": "文字块1",
        "confidence": 0.9950,
        "location": [
          [13, 476],
          [91, 332],
          [125, 351],
          [48, 494]
        ]
      },
      {
        "words": "文字块2",
        "confidence": 0.9910,
        "location": [
          [13, 476],
          [91, 332],
          [125, 351],
          [48, 494]
        ]
      },
      {
        "words": "文字块3",
        "confidence": 0.9910,
        "location": [
          [13, 476],
          [91, 332],
          [125, 351],
          [48, 494]
        ]
      }
    ],
    "extracted_data": {}
  }
}
```

状态码：400

失败响应示例

```
{
  "error_code": "AIS.0103",
}
```

```
} "error_msg": "The image size does not meet the requirements."
```

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入网络图片的base64编码进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

```
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.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

- 传入网络图片的url进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

```
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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

- 传入网络图片的base64编码进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

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

- 传入网络图片的url进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

```
# 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

- 传入网络图片的base64编码进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

```
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)
}
}
```

- 传入网络图片的url进行文字识别，识别过程校验图片倾斜角度，判断待识别字体类型，并校验图片是否包含联系人信息

```
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)
    }
}
```

} }

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见**错误码**。

4.7 护照识别

功能介绍

识别护照首页图片中的文字信息，并以JSON格式返回识别的结构化结果。

当前版本支持2012年及以后发行的中国普通电子护照的全字段识别。中国-港澳台地区、外国护照支持护照下方两行国际化的机读码识别，并可从中提取7个关键字段信息。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-1 护照示例图



约束与限制

- 支持中国大陆护照的全字段识别。
- 支持含有完整机读码的中国-港澳台地区及外国护照识别。
- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式图片。
- 图像各边的像素大小在15px到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中护照首页区域有效占比超过25%，保证护照首页内容及其边缘包含在图像内。
- 支持图像中护照任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的护照长宽比与实际护照相差不超过10%。
- 能处理反光、暗光等干扰的图片但影响识别精度。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

 说明

用户首次使用需要先[开通服务](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

POST /v2/{project_id}/ocr/passport

表 4-63 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-64 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

表 4-65 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	该参数与url二选一。 图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片最短边不小于15px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。
url	否	String	该参数与image二选一。url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片的url路径目前支持： - 公网http/https url - OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 - 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 - 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 - url中不能存在中文字符，若存在，中文需要进行utf8编码。

参数	是否必选	参数类型	说明
country_code	否	String	输入值为护照颁发国的国家码，根据国家码选择对应的护照识别服务。 - 若输入中未选此字段，服务会根据自己识别判断的护照类型匹配相应的护照识别服务。 - 若选择值为“GENERAL”，则选择护照机器码识别。 - 若选择值为“CHN”，则选择中国护照全字段识别。

响应参数

说明

根据识别的结果，可能有不同的HTTP响应状态码（status code）。例如，200表示API调用成功，400表示调用失败，详细的状态码和响应参数说明如下。

状态码： 200

表 4-66 响应 Body 参数

参数	参数类型	描述
result	PassportResult object	识别结果。 调用失败时不返回此字段。 该结果中包含：13个主要字段，由英文表达；extra_info由本地官方语言表达；主要字段的置信度。置信度越大，字段的值越准确。

表 4-67 PassportResult

参数	参数类型	描述
passport_type	String	护照类型。 - P:普通因私护照 - W:外交护照 - G:公务护照 说明 该字段仅限中国大陆护照。
country_code	String	护照签发国的国家码。
passport_number	String	护照号码。

参数	参数类型	描述
nationality	String	护照持有人国籍。 说明 该字段仅限中国大陆护照。
surname	String	姓。
given_name	String	名字。
sex	String	性别。
date_of_birth	String	出生日期。返回值例如，1990-12-12。
date_of_expiry	String	护照有效期。返回值例如，2020-07-08。
date_of_issue	String	护照签发日期。返回值例如，2010-07-09。 说明 该字段仅限中国大陆护照。
place_of_birth	String	出生地。 说明 该字段仅限中国大陆护照。
place_of_issue	String	签发地。 说明 该字段仅限中国大陆护照。
issuing_authority	String	签发机关。 因为各个领事馆签发机关简写未统一，所以服务统一对中国的英文简写处理为“P.R.China”。例如：有的签发机关是“P.R.C”，识别结果均显示“P.R.China”。 说明 该字段仅限中国大陆护照。
confidence	Object	字段的置信度，取值范围0~1。 置信度越大，表示识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。 置信度由算法给出，不直接等价于字段的准确率。
extra_info	Object	参数值默认为空。对于中国护照，“extra_info”的值会包含护照上由汉字描述的字段信息，如姓名、出生地等信息。 说明 该字段仅限中国大陆护照。

状态码： 400

表 4-68 响应 Body 参数

参数	参数类型	说明
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时返回的错误信息。 调用成功时不返回此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，护照识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com或ocr.ap-southeast-1.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/passport”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)
- 如何获取Token请参见[认证鉴权](#)。
- 读取护照图片的base64编码进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/passport
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:

```
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
  "country_code": "GENERAL"
}
```
- 读取护照图片的url进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/passport
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBgkqhkiG...
Request Body:

```
{
  "url": "https://BucketName.obs.xxx.com/ObjectName",
  "country_code": "GENERAL"
}
```

响应示例

状态码：200

中国护照

```
{
  "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": "张三",
        "sex": "女",
        "place_of_birth": "湖南",
        "place_of_issue": "广东",
        "issuing_authority": "xxx出入境管理局",
        "nationality": "中国"
      }
    },
    "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
    }
  }
}

```

外国护照

[illegible]

状态码: 400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 读取护照图片的base64编码进行文字识别

```
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/4RFZRXhpZgAATU0AKgAAAQg...");
        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());
        }
    }
}
```

- 读取护照图片的url进行文字识别

```
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

- 读取护照图片的base64编码进行文字识别

```
# 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/4RFZRXhpZgAATU0AKgAAAQ..."
    )
    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)
```

- 读取护照图片的url进行文字识别

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

- 读取护照图片的base64编码进行文字识别

```
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/4RFZRXhpZgAATU0AKgAAAQ..."
    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)
    }
}
```

- 读取护照图片的url进行文字识别

```
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)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.8 泰文身份证识别

功能介绍

识别泰国身份证中的文字信息，并以JSON格式返回识别的结构化结果。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-2 泰文身份证示例图



约束与限制

- 只支持泰国身份证的识别。
- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式图片。
- 泰国身份证识别图像各边的像素大小在15px到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中身份证区域有效占比超过25%，保证整张身份证内容及其边缘包含在图像内。
- 支持图像中身份证任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的身份证长宽比与实际身份证相差不超过10%。
- 能处理反光、暗光等干扰的图片但影响识别精度。
- 目前只支持识别单张身份证的正面或者反面。

调用方法

请参见[如何调用API](#)。

前提条件

在使用泰文身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-69 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-70 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

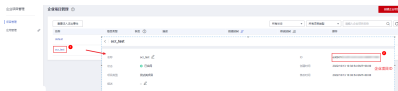
参数	是否必选	参数类型	描述
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 - 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 - 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 - 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-71 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	与url二选一。 图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片最短边不小于15px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。

参数	是否必选	参数类型	说明
url	否	String	与image二选一。 url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片的url路径目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。
side	否	String	标示身份证正面或背面。可选值如下： • front：身份证正面 • back：身份证背面 如果参数值为空或无该参数，系统自动识别，建议填写，准确率更高。
return_portrait_image	否	Boolean	是否返回身份证头像照片的 Base64 编码，可选值如下： • true：返回身份证头像照片的 Base64 编码 • false：不返回身份证头像照片的 Base64 编码 未传入该参数时默认为false，即不返回身份证头像照片的 Base64 编码。
return_portrait_location	否	Boolean	是否返回头像位置坐标，可选值如下： • true：返回身份证头像的位置坐标 • false：不返回身份证头像的位置坐标
return_idcard_type	否	Boolean	是否返回身份证类型，可选值如下： • true: 返回身份证类型，类型包括身份证原件和身份证复印件 • false: 不返回身份证类型

参数	是否必选	参数类型	说明
return_text_location	否	Boolean	识别到的文字块的区域位置信息。可选值包括： true：返回各个文字块区域 false：不返回各个文字块区域 如果无该参数，系统默认不返回文字块区域。如果输入参数不是Boolean类型，则会报非法参数错误。

响应参数

状态码： 200

表 4-72 响应 Body 参数

参数	参数类型	描述
result	ThailandIdcardResult object	识别结果。 调用失败时不返回此字段。

表 4-73 ThailandIdcardResult

参数	参数类型	描述
type	String	返回证件类型，"normal"表示普通泰文证件，"pink"表示外国人身份证件
name_en	String	英文名。
ref_number	String	参考编码。
side	String	标示正面还是反面，取值为front或back。
id_number	String	身份证号。
name_th	String	泰文名字。
first_name_en	String	英文名字。
last_name_en	String	英文姓氏。
date_of_birth_th	String	泰文出生日期。
date_of_birth_en	String	英文出生日期。
religion_th	String	宗教。

参数	参数类型	描述
address_th	String	地址。
date_of_issue_th	String	泰文签发日期。
date_of_issue_en	String	英文签发日期。
date_of_expiry_th	String	泰文有效期。
date_of_expiry_en	String	英文有效期。
serial_number	String	序列号。
card_number	String	身份证反面卡号。
laser_number	String	激光码。
confidence	ThailandIdcardConfidence object	字段的置信度，取值范围0~1。 置信度越大，本次识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于字段的准确率。
portrait_image	String	头像的Base64编码。当输入参数“return_portrait_image”为“true”时，才返回该参数。
portrait_location	Array<Array<Integer>>	头像在原图上的位置。当输入参数“return_portrait_location”为“true”时，才返回该参数。以列表形式显示，包含头像区域四个顶点的二维坐标（x,y），坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向
idcard_type	String	身份证的类型。当输入参数“return_idcard_type”为“true”时，才返回该参数。取值如下所示： • normal：身份证原件 • copy：复印的身份证
text_location	Object	对应所有在原图上识别到的字段位置信息，包含所有文字区域四个顶点的二维坐标（x,y）。采用图像坐标系，坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。

表 4-74 ThailandIdcardConfidence

参数	参数类型	描述
id_number	Float	身份证号置信度。

参数	参数类型	描述
name_th	Float	泰文名字置信度。
name_en	Float	英文名置信度。
ref_number	Float	参考编码置信度。
first_name_en	Float	英文名字置信度。
last_name_en	Float	英文姓氏置信度。
date_of_birth_th	Float	泰文出生日期置信度。返回值例如，5 ธ.พ. 2493。
date_of_birth_en	Float	英文出生日期置信度。返回值例如，5 Feb. 1950。
religion_th	Float	宗教置信度。
address_th	Float	地址置信度。
date_of_issue_th	Float	泰文签发日期置信度。
date_of_issue_en	Float	英文签发日期置信度。
date_of_expir_y_th	Float	泰文有效期置信度。
date_of_expir_y_en	Float	英文有效期置信度。
serial_number	Float	序列号置信度。
card_number	Float	身份证反面卡号置信度。
laser_number	Float	激光码置信度。

状态码： 200

表 4-75 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时返回的错误信息。 调用成功时不返回此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。

例如，泰文身份证识别服务部署在“亚太-曼谷”区域的“endpoint”为“ocr.ap-southeast-2.myhuaweicloud.com”或“ocr.ap-southeast-2.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project_id}/thailand-id-card”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)

- 如何获取Token请参见[构造请求](#)。

- 请求示例（方式一：使用图片的Base64编码）

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

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...

Request Body:

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

- 请求示例（方式二：使用图片URL）

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

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...

Request Body:

```
{
  "url": "https://BucketName.obs.xxx.com/ObjectName"
}
```

- Python3语言请求代码示例（其他语言参照下列示例编写或使用OCR SDK）

encoding:utf-8

import requests

import base64

url = "https://{endpoint}/v2/{project_id}/ocr/thailand-id-card"

token = "用户获取到的实际token值"

headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/thailand-id-card-demo.png' # 读取本地图片

with open(imagepath, "rb") as bin_data:

image_data = bin_data.read()

image_base64 = base64.b64encode(image_data).decode("utf-8") # 使用图片的Base64编码

payload = {"image": image_base64}

response = requests.post(url, headers=headers, json=payload)

print(response.text)

响应示例

状态码：200

成功响应示例（正面）

```
{
  "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 日.月. 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 日.月. 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"
}
}
```

成功响应示例（背面）

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

状态码：400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入泰文身份证图片的base64编码进行文字识别

```
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());
        }
    }
}
```

- 传入泰文身份证图片的url进行文字识别

```
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

- 传入泰文身份证图片的base64编码进行文字识别

```
# 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)
```

- 传入泰文身份证图片的url进行文字识别

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

- 传入泰文身份证图片的base64编码进行文字识别

```
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)
    }
}
```

- 传入泰文身份证图片的url进行文字识别

```
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)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4.9 哥伦比亚身份证识别

功能介绍

识别哥伦比亚身份证中的文字信息，并将识别的结构化结果返回给用户。

约束与限制

- 支持哥伦比亚身份证的识别。
- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式图片。
- 图像各边的像素大小在15px到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中身份证区域有效占比超过80%，保证整张身份证内容及其边缘包含在图像内。
- 支持图像中身份证任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的身份证长宽比与实际身份证相差不超过10%。
- 能处理反光、暗光等干扰的图片但影响识别精度。
- 目前只支持识别单张身份证的正面或者反面。

调用方法

请参见[如何调用API](#)。

前提条件

在使用之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-76 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-77 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“ 企业项目管理 ”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。 企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 • 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 • 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 • 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-78 请求 Body 参数

参数	是否必选	参数类型	描述
image	否	String	该参数与url二选一。 图像数据，base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片最短边不小于15像素，最长边不超过8192像素，支持JPG/PNG/BMP/TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。
url	否	String	与image二选一。url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片的url路径目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限>。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 • url中不能存在中文字符，若存在，中文需要进行utf8编码。

响应参数

状态码： 200

表 4-79 响应 Body 参数

参数	参数类型	描述
result	表4-80 object	调用成功时表示调用结果。调用失败时无此字段。

表 4-80 ColombialdCardResult

参数	参数类型	描述
side	String	证件图片正反面信息。可选值包括： - front: 证件图片正面 - back: 证件图片反面
number	String	卡证编号。当响应字段"side"为front时，返回此字段。
name	String	名。当响应字段"side"为front时，返回此字段。
last_name	String	姓。当响应字段"side"为front时，返回此字段。
birth_date	String	出生日期。
birth_place	String	出生地。
gender	String	性别。
blood_type	String	血型。
issue_date	String	签发日期。
issue_authority	String	签发地区。
height	String	身高。
citizen_code1	String	公民编码一。
citizen_code2	String	公民编码二。
citizen_code3	String	公民编码三。
confidence	Map<String,Number>	相关字段的置信度信息，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。注：置信度由算法给出，不直接等价于对应字段的准确率。

状态码： 400

表 4-81 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

请求示例

说明

- Endpoint即调用API的请求地址，不同服务不同区域的Endpoint不同，具体请参见[终端节点](#)。
例如，哥伦比亚身份证识别服务部署在“拉美-墨西哥城二”区域的“endpoint”为“ocr.la-north-2.myhuaweicloud.com”，请求URL为“https://ocr.la-north-2.myhuaweicloud.com/v2/{project_id}/ocr/colombia-id-card”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token请参见[构造请求](#)。
- 传入哥伦比亚身份证图片的base64编码进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/colombia-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:

```
{
  "image" : "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
}
```
- 传入哥伦比亚身份证图片的url进行文字识别
POST https://{endpoint}/v2/{project_id}/ocr/colombia-id-card
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:

```
{
  "url" : "https://BucketName.obs.myhuaweicloud.com/ObjectName"
}
```

响应示例

状态码： 200

成功响应示例

```
{
  "result" : {
    "side" : "front",
    "number" : "1.039xxxx",
    "name" : "VALENxxx",
    "last_name" : "SANCHxxxx",
    "confidence" : {
      "number" : 0.9888,
      "name" : 0.9969,
      "last_name" : 0.9959
    }
  }
}
```

```
}  
}
```

状态码： 400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入哥伦比亚身份证图片的base64编码进行文字识别

```
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 RecognizeColombialdCardSolution {  
  
    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();  
        RecognizeColombialdCardRequest request = new RecognizeColombialdCardRequest();  
        ColombialdCardRequestBody body = new ColombialdCardRequestBody();  
        body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");  
        request.withBody(body);  
        try {  
            RecognizeColombialdCardResponse response = client.recognizeColombialdCard(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.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

- 传入哥伦比亚身份证图片的url进行文字识别

```
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 RecognizeColombialdCardSolution {

    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();
        RecognizeColombialdCardRequest request = new RecognizeColombialdCardRequest();
        ColombialdCardRequestBody body = new ColombialdCardRequestBody();
        body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
        request.withBody(body);
        try {
            RecognizeColombialdCardResponse response = client.recognizeColombialdCard(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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 传入哥伦比亚身份证图片的base64编码进行文字识别

```
# 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 = RecognizeColombialdCardRequest()
        request.body = ColombialdCardRequestBody(
            image="/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
        )
        response = client.recognize_colombia_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)
```

- 传入哥伦比亚身份证图片的url进行文字识别

```
# 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 = RecognizeColombialdCardRequest()
        request.body = ColombialdCardRequestBody(
            url="https://BucketName.obs.myhuaweicloud.com/ObjectName"
        )
        response = client.recognize_colombia_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

- 传入哥伦比亚身份证图片的base64编码进行文字识别

```
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.RecognizeColombialdCardRequest{}
    imageColombialdCardBody := "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAA..."
    request.Body = &model.ColombialdCardBody{
        Image: &imageColombialdCardBody,
    }
    response, err := client.RecognizeColombialdCard(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 传入哥伦比亚身份证图片的url进行文字识别

```
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.RecognizeColombialdCardRequest{
    urlColombialdCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
    request.Body = &model.ColombialdCardRequestBody{
        Url: &urlColombialdCardRequestBody,
    }
}
response, err := client.RecognizeColombialdCard(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例
400	失败响应示例

状态码请参见[状态码](#)。

错误码

请参见[错误码](#)。

4.10 柬文身份证识别

功能介绍

识别柬文身份证图片中的文字内容，并将识别的结构化结果返回给用户。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

约束与限制

- 目前只支持单张柬文身份证正面识别。

- 只支持识别PNG、JPG、BMP、TIFF格式的图片。单个图对应的Base64编码不超过10MB。
- 图像各边的像素大小在15到8192px之间。
- 支持图像中柬文身份证任意角度的水平旋转。
- 能处理反光、暗光等干扰的图片但影响识别精度。

调用方法

请参见[如何调用API](#)。

前提条件

在使用柬文身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-82 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。

请求参数

表 4-83 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

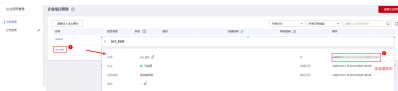
参数	是否必选	参数类型	描述
Enterprise-Project-Id	否	String	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 - 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 - 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 - 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

表 4-84 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	与url二选一。单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片尺寸不小于15像素，最长边不超过8192像素，支持JPG/PNG/BMP/TIF格式。

参数	是否必选	参数类型	说明
url	否	String	与image二选一。url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片的url路径，目前支持： • 公网http/https url • OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 • 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 • 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。 • url中不能存在中文字符，若存在，中文需要进行utf8编码。
return_portrait_image	否	Boolean	是否返回头像内容开关，可选值如下所示： • true: 返回身份证头像照片的 base64 编码 • false: 不返回身份证头像照片的 base64 编码 未传入该参数时默认为“false”，即不返回身份证头像照片的 base64 编码。
return_portrait_location	否	Boolean	是否返回头像坐标的开关，可选值如下所示： • true: 返回身份证头像的位置坐标 • false: 不返回身份证头像的位置坐标 未传入该参数时默认为“false”，即不返回身份证的头像坐标。
return_idcard_type	否	Boolean	是否返回身份证类型的开关，可选值如下所示： • true:返回身份证的类型，类型包括身份证原件以及身份证复印件 • false: 不返回身份证的类型
detect_border_integrity	否	Boolean	返回身份证边框完整性的告警结果的开关，可选值如下所示： • true: 打开身份证图像边框完整性告警功能 • false: 关闭身份证图像边框完整性告警功能

参数	是否必选	参数类型	说明
detect_blocking_within_border	否	Boolean	返回身份证内部是否有被遮挡的告警结果的开关，可选值如下所示： • true：打开身份证内部是否有被遮挡的告警功能 • false：关闭身份证内部是否有被遮挡的告警功能
detect_blur	否	Boolean	返回身份证模糊告警结果的开关，可选值如下所示： • true:打开身份证是否模糊的告警功能 • false：关闭身份证是否模糊的告警功能
detect_glare	否	Boolean	返回身份证是否反光的告警结果的开关，可选值如下所示： • true：打开身份证是否反光的告警功能 • false：关闭身份证是否反光的告警功能
return_adjusted_image	否	Boolean	返回身份证四点原图的base64编码 • true: 返回身份证原图的base64编码 • false：不返回身份证原图的base64编码
detect_tampering	否	Boolean	返回身份证人像是否被篡改的告警结果的开关，可选值如下所示： • true: 打开身份证人像是否被篡改的告警功能 • false：关闭身份证人像被篡改的告警功能 不支持精细化的P图
detect_reproduce	否	Boolean	返回判断身份证图像是否经过翻拍告警的开关，可选值如下所示： • true：打开判断身份证图像是否经过翻拍告警的功能 • false：关闭判断身份证图像是否经过翻拍告警的功能

响应参数

状态码： 200

表 4-85 响应 Body 参数

参数	参数类型	描述
result	CambodianIdCardResult object	识别结果。 调用失败时此字段为空。

表 4-86 CambodianIdCardResult

参数	参数类型	描述
id_number	String	身份证号码。
name_kh	String	高棉语版姓名。
name_en	String	英文姓名。
birth_date	String	出生日期。
sex	String	性别。
height	String	身高。
birth_place	String	出生地。
address	String	地址，以空格分隔。
issue_date	String	签发起始日期。
expiry_date	String	签发结束日期。
description	String	图片中的个人特征。
machine_code 1	String	机器码第一行。
machine_code 2	String	机器码第二行。
machine_code 3	String	机器码第三行。
portrait_image	String	头像的base64编码。当输入参数“return_portrait_image”为“true”时，才返回该参数。
portrait_location	Array<Array<Integer>>	头像在原图上的位置。当输入参数“return_portrait_location”为“true”时，才返回该参数。以列表形式显示，包含头像区域四个顶点的二维坐标（x,y），坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。

参数	参数类型	描述
idcard_type	String	身份证的类型。当输入参数 “ idcard_type ” 为 “ true ” 时，才返回该参数。取值如下所示： • normal: 身份证原件 • copy: 复印的身份证
adjusted_image	String	身份证原图的base64编码。 当输入参数 "return_adjusted_image"为"true"时，才返回该参数。
detect_border_integrity_result	Boolean	身份证图片边框完整性告警结果， "true"表示边框不完整， "false"表示边框完整。仅在输入参数 detect_border_integrity为true时， 返回该字段。
detect_blocking_within_border_result	Boolean	身份证图像框内是否存在遮挡的告警结果， "true"表示边框内部存在遮挡， "false"表示边框内部完整。仅在输入参数 detect_blocking_within_border为true时， 返回该字段。
detect_blur_result	Boolean	身份证模糊告警结果， "true"表示图片模糊， "false"表示身份证清晰。仅在输入参数 detect_blur为true时， 返回该字段。
detect_glare_result	Boolean	身份证反光告警结果， "true"表示身份证反光， "false"表示是身份证无反光。仅在输入参数 detect_glare为true时， 返回该字段。
detect_tampering_result	Boolean	身份证人像被篡改的告警结果， "true"表示身份证人像被篡改， "false"表示是身份证人像未被篡改。仅在输入参数detect_tampering为true时， 返回该字段。
detect_reproduce_result	Boolean	身份证是否经过翻拍的告警结果， “ true ” 表示身份证经过翻拍， “ false ” 表示身份证未经过翻拍。仅在输入参数detect_reproduce为true时， 返回该字段。
score_info	CambodianIdCardScoreInformationResult object	告警分数，包含字段： idcard_type_score、border_integrity_score、blocking_within_border_score、blur_score、glare_score、tampering_score、reproduce_score以上字段取值范围为[0, 99]。
confidence	Object	字段的置信度，取值范围0~1。 置信度越大，本次识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。 置信度由算法给出，不直接等价于字段的准确率。

表 4-87 CambodianIdCardScoreInformationResult

参数	参数类型	描述
idcard_type_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示复印件，小于等于50表示原件，值越靠近99，表示复印件的可能性越大，值越靠近0，表示原件的可能性越大。仅在传入参数return_idcard_type为true时，返回该字段。
border_integrity_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示边框不完整，小于50表示边框完整，值越靠近99，表示边框不完整的可能性越大，值越靠近0，表示边框完整的可能性越大。仅在传入参数detect_border_integrity为true时，返回该字段。
blocking_within_border_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示框内有遮挡，小于50表示框内无遮挡，值越靠近99，表示框内有遮挡的可能性越大，值越靠近0，表示框内无遮挡的可能性越大。仅在传入参数detect_blocking_within_border为true时，返回该字段。
blur_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示身份证模糊，小于50表示身份证清晰，值越靠近99，表示身份证模糊的可能性越大，值越靠近0，表示身份证清晰的可能性越大。仅在传入参数detect_blur为true时，返回该字段。
glare_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示身份证反光，小于50表示身份证不反光，值越靠近99，表示身份证反光的可能性越大，值越靠近0，表示身份证不反光的可能性越大。仅在传入参数detect_glare为true时，返回该字段。
tampering_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示身份证人像被其他非身份证人像篡改过，小于50表示身份证人像未被篡改，值越靠近99，表示身份证人像被篡改的可能性越大，值越靠近0，表示身份证未人像被篡改的可能性越大。仅在传入参数detect_tampering为true时，返回该字段。
reproduce_score	Integer	告警分数，字段取值范围[0, 99]值大于50表示身份证经过翻拍，小于50表示身份证未经过翻拍，值越靠近99，表示身份证图像被翻拍过的可能性越大，值越靠近0，表示身份证图像未被翻拍的可能性越大。仅在传入参数detect_reproduce为true时，返回该字段。

状态码： 400


```
    "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
    ]
  ]
}
```

状态码：400

失败响应示例

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

SDK 代码示例

SDK代码示例如下。

说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

Java

- 传入柬文身份证图片的base64编码进行文字识别

```
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());
    }
}
```

- 传入柬文身份证图片的url进行文字识别

```
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

- 传入柬文身份证图片的base64编码进行文字识别

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)
```

- 传入柬文身份证图片的url进行文字识别

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 = CambodianIdCardBody(
            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

- 传入柬文身份证图片的base64编码进行文字识别

```
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{
        imageCambodianIdCardBody: "9j/4AAQSkZJRgABAQgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
    }
```

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

- 传入柬文身份证图片的url进行文字识别

```
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)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	成功响应示例

状态码	描述
400	失败响应示例

状态码请参见[状态码](#)。

错误码

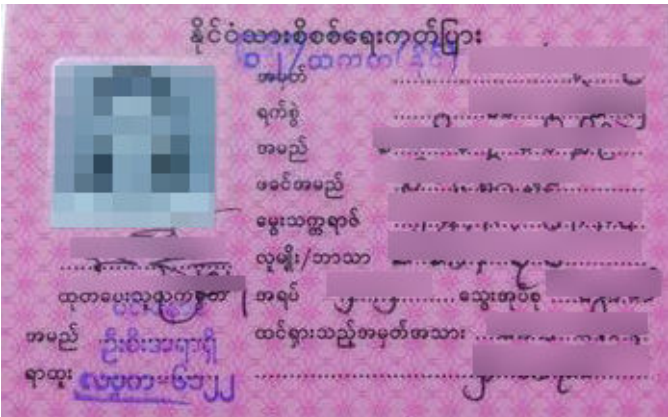
错误码请参见[错误码](#)。

4.11 缅文身份证识别

功能介绍

识别缅甸身份证中的文字信息，并将识别的结构化结果返回给用户。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-3 缅甸文身份证示例图



约束与限制

- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式的图片。
- 缅甸身份证识别图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中身份证区域有效占比超过25%，保证整张身份证内容及其边缘包含在图像内。
- 支持图像中身份证任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的身份证长宽比与实际身份证相差不超过10%。
- 能处理反光、暗光等干扰的图片但影响识别精度。
- 目前只支持识别单张身份证的正面或者反面。

调用方法

请参见[如何调用API](#)。

前提条件

在使用缅甸身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

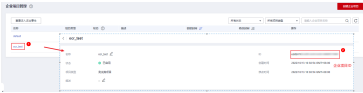
说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

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

表 4-89 路径参数

参数	是否必选	说明
endpoint	是	终端节点，即调用API的请求地址。 不同服务不同区域的endpoint不同，您可以从 终端节点 中获取。
project_id	是	项目ID，您可以从 获取项目ID 中获取。
Enterprise-Project-Id	否	企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。 获取方法：进入“企业项目管理”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。  企业项目创建步骤请参见用户指南。 说明 创建企业项目后，在传参时，有以下三类场景。 - 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。 - 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。 - 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。

请求参数

表 4-90 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用于获取操作API的权限。 获取Token接口 响应消息头中X-Subject-Token的值即为Token。
Content-Type	是	String	发送的实体的MIME类型，参数值为“application/json”。

表 4-91 请求 Body 参数

参数	是否必选	参数类型	说明
image	否	String	与url二选一。 图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。 图片最短边不小于15px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF格式。 图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。
url	否	String	与image二选一。 图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持： - 公网http/https url - OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见配置OBS访问权限。 说明 - 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。 - 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。

参数	是否必选	参数类型	说明
convert_unicode	否	Boolean	输出格式，可选值如下： - true：输出为unicode格式 - false：输出为zawgyi格式 如果参数值为空或无该参数，默认输出为zawgyi格式。
return_confidence	否	Boolean	是否返回置信度，可选值如下： true：返回置信度 false：不返回置信度 如果无该参数，系统默认不返回置信度。如果输入参数不是Boolean类型，则会报非法参数错误。
return_portrait_image	否	Boolean	是否返回身份证头像照片的 Base64 编码，可选值如下： - true：返回身份证头像照片的 Base64 编码 - false：不返回身份证头像照片的 Base64 编码 未传入该参数时默认为false，即不返回身份证头像照片的 Base64 编码。
return_portrait_location	否	Boolean	是否返回身份证头像的位置，可选值如下： - true：返回身份证头像的位置 - false：不返回身份证头像的位置
return_idcard_type	否	Boolean	是否返回身份证类型，可选值如下所示： - true: 返回身份证类型，类型包括身份证原件和身份证复印件 - false: 不返回身份证类型
return_translation	否	Boolean	是否返回转译信息的开关，可选值如下所示： - true: 返回转译信息。 - false: 不返回转译信息。

响应参数

状态码： 200

表 4-92 响应 Body 参数

参数	参数类型	描述
result	MyanmarIdcardResult object	调用结果。 调用失败时不返回此字段。

表 4-93 MyanmarIdcardResult

参数	参数类型	描述
side	String	标示身份证是正面还是反面，取值为front或back。
class	String	身份证类型。取值如下： - new_version：新版身份证 - old_version：旧版身份证
nrc_id	String	身份证号码。
issue_date	String	签发日期。
name	String	姓名。
father_name	String	父亲名字。
birth	String	出生日期。
bloodlines_religion	String	族群或宗教。
height	String	身高。
blood_group	String	血型。
card_id	String	身份证的卡号（背面）。
nrc_id_back	String	背面的身份证号码。
profession	String	职业。
address	String	地址。
confidence	MyanmarIdcardConfidence object	字段的置信度，取值范围0~1。 置信度越大，本次识别的字的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于字段的准确率。
portrait_image	String	头像的Base64编码。当输入参数“return_portrait_image”为“true”时，才返回该参数。

参数	参数类型	描述
portrait_location	Array<Array<Integer>>	头像在原图上的位置。当输入参数“return_portrait_location”为“true”时，才返回该参数。以列表形式显示，包含头像区域四个顶点的二维坐标（x,y），坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。
idcard_type	String	身份证的类型。当输入参数“return_idcard_type”为“true”时，才返回该参数。取值如下所示： • normal：身份证原件 • copy：身份证复印件
translation_info	MyanmarIdcardTranslationInfo object	转译信息。仅当输入参数return_translation为true时，返回该字段。该字段包含name_translation、nrc_id_translation的信息。当无法找到合适的翻译字段时，保留原始缅文字符。

表 4-94 MyanmarIdcardConfidence

参数	参数类型	描述
nrc_id	Float	身份证号码置信度。
issue_date	Float	签发日期置信度。
name	Float	姓名置信度。
birth	Float	出生日期置信度。
bloodlines_religion	Float	族群或宗教置信度。
height	Float	身高置信度。
blood_group	Float	血型置信度。
card_id	Float	身份证卡号（背面）置信度。
nrc_id_back	Float	背面的身份证号。
profession	Float	职业置信度。
address	Float	地址置信度。

表 4-95 MyanmarIdcardTranslationInfo

参数	参数类型	描述
name_translation	String	名字转译。仅当输入参数return_translation为true时，返回该字段。
father_name_translation	String	父亲名字的转译。仅当输入参数return_translation为true时，返回该字段。
nrc_id_translation	String	身份证号码转译。仅当输入参数return_translation为true时，返回该字段。
birth_translation	String	出生日期转译。仅当输入参数return_translation为true时，返回该字段。

状态码： 400

表 4-96 响应 Body 参数

参数	参数类型	描述
error_code	String	调用失败时的错误码，具体请参见 错误码 。 调用成功时不返回此字段。
error_msg	String	调用失败时返回的错误信息。 调用成功时不返回此字段。

请求示例

说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。
例如，缅文身份证识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com”或“ocr.ap-southeast-1.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project_id}/ocr/myanmar-id-card”，“project_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token请参见[构造请求](#)。
- 请求示例（方式一：使用图片的Base64编码）
POST https://{endpoint}/v2/{project_id}/ocr/myanmar-id-card

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

```
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAABwESAAAd...",
  "convert_unicode": true,
  "return_confidence": true,
  "return_portrait_image": true,
  "return_portrait_location": true,
```

- ```
 "return_idcard_type": true
 }
}
```
- 请求示例（方式二：使用图片URL）  
POST https://{endpoint}/v2/{project\_id}/ocr/myanmar-id-card  
  
Request Header:  
Content-Type: application/json  
X-Auth-Token:  
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgggVBgkqhkiG...  
Request Body:  
{  
 "url": "https://BucketName.obs.xxx.com/ObjectName",  
 "convert\_unicode": true,  
 "return\_confidence": true,  
 "return\_portrait\_image": true,  
 "return\_portrait\_location": true,  
 "return\_idcard\_type": true  
}
  - Python3语言请求代码示例（其他语言参照下列示例编写或使用OCR SDK）  
# encoding:utf-8  
  
import requests  
import base64  
  
url = "https://{endpoint}/v2/{project\_id}/ocr/myanmar-id-card"  
token = "用户获取到的实际token值"  
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}  
  
imagepath = r'./data/myanmar-id-card-demo.png' # 读取本地图片  
with open(imagepath, "rb") as bin\_data:  
 image\_data = bin\_data.read()  
image\_base64 = base64.b64encode(image\_data).decode("utf-8") # 使用图片的Base64编码  
payload = {"image": image\_base64}  
response = requests.post(url, headers=headers, json=payload)  
print(response.text)

## 响应示例

状态码：200

成功响应示例（正面）

```
{
 "result": {
 "side": "front",
 "nrc_id": "XXXXXXXX",
 "issue_date": "00-0-0000",
 "name": "XXX",
 "father_name": "XXX",
 "birth": "00-00-0000",
 "bloodlines_religion": "00000000",
 "height": "000",
 "blood_group": "00",
 "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"
}
}
```

#### 成功响应示例（背面）

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

#### 状态码：400

#### 失败响应示例

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

## SDK 代码示例

SDK代码示例如下。

### 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

## Java

- 传入缅文身份证图片的base64编码进行文字识别

```
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());
}
}
```

- 传入缅文身份证图片的url进行文字识别

```
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

- 传入缅文身份证图片的base64编码进行文字识别

# 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)
```

- 传入缅文身份证图片的url进行文字识别

# 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

- 传入缅文身份证图片的base64编码进行文字识别

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)
}
}
```

- 传入缅文身份证图片的url进行文字识别

```
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)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.12 缅文驾驶证识别

### 功能介绍

识别缅甸驾驶证中的文字信息，并以JSON格式返回识别的结构化结果。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-4 缅文驾驶证示例图



约束与限制

- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式的图片。
- 图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 目前只支持单张缅文驾驶证正面识别。
- 支持图像中缅文驾驶证任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的驾驶证长宽比与实际驾驶证相差不超过10%。
- 能处理反光、暗光等干扰的图片但影响识别精度。

调用方法

请参见[如何调用API](#)。

前提条件

在使用缅文驾驶证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

URI

POST /v2/{project\_id}/ocr/myanmar-driver-license

表 4-97 路径参数

| 参数       | 是否必选 | 说明                                                                      |
|----------|------|-------------------------------------------------------------------------|
| endpoint | 是    | 终端节点，即调用API的请求地址。<br>不同服务不同区域的endpoint不同，您可以从 <a href="#">终端节点</a> 中获取。 |

| 参数         | 是否必选 | 说明                                    |
|------------|------|---------------------------------------|
| project_id | 是    | 项目ID，您可以从 <a href="#">获取项目ID</a> 中获取。 |

请求参数

表 4-98 请求 Header 参数

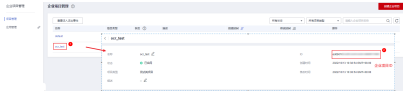
| 参数                    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X-Auth-Token          | 是    | String | 用户Token。<br>用于获取操作API的权限。 <a href="#">获取Token接口</a> 响应消息头中X-Subject-Token的值即为Token。                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Content-Type          | 是    | String | 发送的实体的MIME类型，参数值为“application/json”。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Enterprise-Project-Id | 否    | String | <p>企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。</p> <p>获取方法：进入“<a href="#">企业项目管理</a>”页面，单击企业项目名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。</p>  <p>企业项目创建步骤请参见用户指南。</p> <p><b>说明</b><br/>创建企业项目后，在传参时，有以下三类场景。</p> <ul style="list-style-type: none"><li>• 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。</li><li>• 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。</li><li>• 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。</li></ul> |

表 4-99 请求 Body 参数

| 参数              | 是否必选 | 参数类型    | 说明                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| image           | 否    | String  | 与url二选一。<br>图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。<br>图片最短边不小于15px，最长边不超过4096px，支持JPEG、JPG、PNG、BMP、TIFF格式。<br>图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。                                                                                                                                                               |
| url             | 否    | String  | 与image二选一。<br>图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持： <ul style="list-style-type: none"><li>公网http/https url</li><li>OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见<a href="#">配置OBS访问权限</a>。</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li><li>请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li></ul> |
| convert_unicode | 否    | Boolean | 输出格式。可选值如下： <ul style="list-style-type: none"><li>true：输出为unicode格式。</li><li>false：输出为zawgyi格式。</li></ul> 如果参数值为空或无该参数，默认输出为zawgyi格式。                                                                                                                                                                                                                                                                   |

响应参数

状态码： 200

表 4-100 响应 Body 参数

| 参数     | 参数类型                              | 描述                    |
|--------|-----------------------------------|-----------------------|
| result | MyanmarDriverLicenseResult object | 调用结果。<br>调用失败时不返回此字段。 |

表 4-101 MyanmarDriverLicenseResult

| 参数              | 参数类型                                  | 描述                                                                               |
|-----------------|---------------------------------------|----------------------------------------------------------------------------------|
| card_number     | String                                | 缅文驾驶证号。                                                                          |
| card_number_en  | String                                | 英文驾驶证号。                                                                          |
| name            | String                                | 缅文名字。                                                                            |
| name_en         | String                                | 英文名字。                                                                            |
| nrc_id          | String                                | 缅文nrc号码。                                                                         |
| nrc_id_en       | String                                | 英文nrc号码。                                                                         |
| birth           | String                                | 缅文出生日期。                                                                          |
| birth_en        | String                                | 英文出生日期。                                                                          |
| blood_group     | String                                | 缅文血型。                                                                            |
| blood_group_en  | String                                | 英文血型。                                                                            |
| expired_date    | String                                | 缅文有效期。                                                                           |
| expired_date_en | String                                | 英文有效期。                                                                           |
| confidence      | MyanmarDriverLicenseConfidence object | 字段的置信度，取值范围0~1。<br>置信度越大，本次识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于字段的准确率。 |

表 4-102 MyanmarDriverLicenseConfidence

| 参数             | 参数类型  | 描述         |
|----------------|-------|------------|
| card_number    | Float | 缅文驾驶证号置信度。 |
| card_number_en | Float | 英文驾驶证号置信度。 |
| name           | Float | 缅文名字置信度。   |

| 参数              | 参数类型  | 描述          |
|-----------------|-------|-------------|
| name_en         | Float | 英文名字置信度。    |
| nrc_id          | Float | 缅文nrc号码置信度。 |
| nrc_id_en       | Float | 英文nrc号码置信度。 |
| birth           | Float | 缅文出生日期置信度。  |
| birth_en        | Float | 英文出生日期置信度。  |
| blood_group     | Float | 缅文血型置信度。    |
| blood_group_en  | Float | 英文血型置信度。    |
| expired_date    | Float | 缅文有效期置信度。   |
| expired_date_en | Float | 英文有效期置信度。   |

状态码： 400

表 4-103 响应 Body 参数

| 参数         | 参数类型   | 描述                                                    |
|------------|--------|-------------------------------------------------------|
| error_code | String | 调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。<br>调用成功时不返回此字段。 |
| error_msg  | String | 调用失败时返回的错误信息。<br>调用成功时不返回此字段。                         |

## 请求示例

### 说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。  
例如，缅文驾驶证识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com”或“ocr.ap-southeast-1.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project\_id}/ocr/myanmar-driver-license”，“project\_id”为项目ID，获取方法请参见[获取项目ID](#)
- 如何获取Token请参见[构造请求](#)。
- 请求样例（方式一：使用图片的Base64编码）  
POST https://{endpoint}/v2/{project\_id}/ocr/myanmar-driver-license  
  
Request Header:  
Content-Type: application/json  
X-Auth-Token:  
MIINRwYJKoZIhvcNAQcColINODCCDTQCAQExDTALBgIghkgBZQMEEAgEwgguVBgkqhkiG...  
Request Body:  
{

```
"image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
"convert_unicode": true
}
```

- 请求样例（方式二：使用图片URL）

POST https://{endpoint}/v2/{project\_id}/ocr/myanmar-driver-license

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZlIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgggVBGkqhkiG...

Request Body:

```
{
 "url": "https://BucketName.obs.xxx.com/ObjectName",
 "convert_unicode": true
}
```

- Python3语言请求代码示例（其他语言参照下列示例编写或使用OCR SDK）

```
encoding:utf-8
```

```
import requests
```

```
import base64
```

```
url = "https://{endpoint}/v2/{project_id}/ocr/myanmar-driver-license"
```

```
token = "用户获取得到的实际token值"
```

```
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}
```

```
imagepath = r'./data/myanmar-driver-license-demo.png' # 读取本地图片
```

```
with open(imagepath, "rb") as bin_data:
```

```
 image_data = bin_data.read()
```

```
image_base64 = base64.b64encode(image_data).decode("utf-8") # 使用图片的Base64编码
```

```
payload = {"image": image_base64}
```

```
response = requests.post(url, headers=headers, json=payload)
```

```
print(response.text)
```

## 响应示例

状态码：200

成功响应示例

```
{
 "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",
 "expired_date": "□□-□-□□□□",
 "expired_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,
 "expired_date": 0.9994,
 "expired_date_en": 0.758
 }
 }
}
```

```
}
}
```

**状态码：400**

失败响应示例

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

## SDK 代码示例

SDK代码示例如下。

### 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

## Java

- 传入缅文驾驶证图片的base64编码进行文字识别

```
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());
 }
}
}
```

- 传入缅甸驾驶证图片的url进行文字识别

```
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

- 传入缅文驾驶证图片的base64编码进行文字识别

```
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)
```

- 传入缅文驾驶证图片的url进行文字识别

```
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

- 传入缅文驾驶证图片的base64编码进行文字识别

```
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)
 }
}
```

- 传入缅文驾驶证图片的url进行文字识别

```
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)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.13 智利身份证识别

### 功能介绍

识别智利身份证图片中的文字内容，并将识别的结果以JSON格式返回给用户。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-5 智利身份证示例图



### 约束与限制

- 只支持识别PNG、JPG、BMP、TIFF格式的图片。
- 图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 目前只支持单张智利身份证正面识别。
- 支持图像中智利身份证任意角度的水平旋转。
- 能处理反光、暗光等干扰的图片但影响识别精度。

### 调用方法

请参见[如何调用API](#)。

### 前提条件

在使用智利身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

#### 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

POST /v2/{project\_id}/ocr/chile-id-card

表 4-104 路径参数

| 参数         | 是否必选 | 说明                                                                                                                              |
|------------|------|---------------------------------------------------------------------------------------------------------------------------------|
| endpoint   | 是    | 终端节点，即调用API的请求地址。<br>不同服务不同区域的endpoint不同，您可以从 <a href="#">终端节点</a> 中获取。<br>智利身份证识别接口的endpoint为ocr.la-south-2.myhuaweicloud.com。 |
| project_id | 是    | 项目ID，您可以从 <a href="#">获取项目ID</a> 中获取。                                                                                           |

请求参数

表 4-105 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                  |
|--------------|------|--------|-------------------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>用于获取操作API的权限。 <a href="#">获取Token接口</a> 响应消息头中X-Subject-Token的值即为Token。 |
| Content-Type | 是    | String | 发送的实体的MIME类型，参数值为“application/json”。                                                |

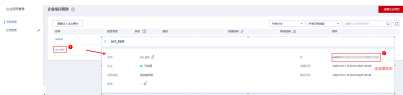
| 参数                    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enterprise-Project-Id | 否    | String | <p>企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。</p> <p>获取方法：进入“<a href="#">企业项目管理</a>”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。</p>  <p>企业项目创建步骤请参见用户指南。</p> <p><b>说明</b></p> <p>创建企业项目后，在传参时，有以下三类场景。</p> <ul style="list-style-type: none"><li>• 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。</li><li>• 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。</li><li>• 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。</li></ul> |

表 4-106 请求 Body 参数

| 参数    | 是否必选 | 参数类型   | 说明                                                                                                                                                                                                                                                |
|-------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| image | 否    | String | <p>与url二选一。</p> <p>图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片尺寸不小于15×15像素，最长边不超过8192像素，支持JPG/PNG/BMP/TIFF格式。</p> <p>图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。</p> |

| 参数  | 是否必选 | 参数类型   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| url | 否    | String | <p>与image二选一。</p> <p>图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持：</p> <ul style="list-style-type: none"><li>• 公网http/https url</li><li>• OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见<a href="#">配置OBS访问权限</a>。</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li><li>• 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li></ul> |

响应参数

状态码： 200

表 4-107 响应 Body 参数

| 参数     | 参数类型                                     | 描述                    |
|--------|------------------------------------------|-----------------------|
| result | <a href="#">ChileIdCardResult</a> object | 调用结果。<br>调用失败时不返回此字段。 |

表 4-108 ChileIdCardResult

| 参数          | 参数类型             | 描述   |
|-------------|------------------|------|
| surname     | Array of strings | 姓氏。  |
| given_name  | String           | 名。   |
| nationality | String           | 国籍。  |
| sex         | String           | 性别。  |
| birth       | String           | 出生日。 |
| issue_date  | String           | 发行日。 |
| expiry_date | String           | 有效期。 |

| 参数              | 参数类型                         | 描述                                                                               |
|-----------------|------------------------------|----------------------------------------------------------------------------------|
| document_number | String                       | 文档编号。                                                                            |
| number          | String                       | 身份证号。                                                                            |
| confidence      | ChileIdCardConfidence object | 字段的置信度，取值范围0~1。<br>置信度越大，本次识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。置信度由算法给出，不直接等价于字段的准确率。 |

表 4-109 ChileIdCardConfidence

| 参数              | 参数类型  | 描述       |
|-----------------|-------|----------|
| surname         | Float | 姓氏置信度。   |
| given_name      | Float | 名置信度。    |
| nationality     | Float | 国籍置信度。   |
| sex             | Float | 性别置信度。   |
| birth           | Float | 出生日置信度。  |
| issue_date      | Float | 发行日置信度。  |
| expiry_date     | Float | 有效期置信度。  |
| document_number | Float | 文档编号置信度。 |
| number          | Float | 身份证号置信度。 |

状态码： 400

表 4-110 响应 Body 参数

| 参数         | 参数类型   | 描述                                                    |
|------------|--------|-------------------------------------------------------|
| error_code | String | 调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。<br>调用成功时不返回此字段。 |
| error_msg  | String | 调用失败时返回的错误信息。<br>调用成功时不返回此字段。                         |

## 请求示例

### 说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。

例如，智利身份证识别服务部署在“拉美-圣地亚哥”区域的“endpoint”为“ocr.la-south-2.myhuaweicloud.com”或“ocr.la-south-2.myhuaweicloud.cn”。请求URL为“https://ocr.la-south-2.myhuaweicloud.com/v2/{project\_id}/ocr/chile-id-card”，“project\_id”为项目ID，获取方法请参见[获取项目ID](#)。

- 如何获取Token请参见[构造请求](#)。

- 请求示例（方式一：使用图片的Base64编码）

POST https://ocr.la-south-2.myhuaweicloud.com/v2/{project\_id}/ocr/chile-id-card

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBgkqhkiG...

Request Body:

```
{
 "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgA..."
}
```

- 请求示例（方式二：使用图片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:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBgkqhkiG...

Request Body:

```
{
 "url": "https://BucketName.obs.xxx.com/ObjectName"
}
```

- Python3语言请求代码示例（其他语言参照下列示例编写或使用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 = "用户获取到的实际token值"

headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'./data/chile-id-card-demo.png' # 读取本地图片

with open(imagepath, "rb") as bin\_data:

image\_data = bin\_data.read()

image\_base64 = base64.b64encode(image\_data).decode("utf-8") # 使用图片的Base64编码

payload = {"image": image\_base64}

response = requests.post(url, headers=headers, json=payload)

print(response.text)

## 响应示例

状态码：200

成功响应示例

```
{
 "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
}
}
```

**状态码：400**

失败响应示例

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

## SDK 代码示例

SDK代码示例如下。

### 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

## Java

- 传入智利身份证图片的base64编码进行文字识别

```
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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrMsg());
}
}
```

- 传入智利身份证图片的url进行文字识别

```
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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
```

## Python

- 传入智利身份证图片的base64编码进行文字识别

# 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 = ChileIdCardBody(
 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)
```

- 传入智利身份证图片的url进行文字识别

# 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

- 传入智利身份证图片的base64编码进行文字识别

```
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)
 }
}
```

- 传入智利身份证图片的url进行文字识别

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

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.14 越南身份证识别

### 功能介绍

识别越南身份证中的文字信息，并将识别的结构化结果返回给用户。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

### 约束与限制

- 只支持越南身份证的识别。
- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式的图片。
- 图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 支持图像中越南身份证任意角度的水平旋转。
- 能处理反光、暗光、防伪标识等干扰的图片但影响识别精度。

### 调用方法

请参见[如何调用API](#)。

### 前提条件

在使用越南身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

POST /v2/{project\_id}/ocr/vietnam-id-card

表 4-111 路径参数

| 参数         | 是否必选 | 说明                                                                      |
|------------|------|-------------------------------------------------------------------------|
| endpoint   | 是    | 终端节点，即调用API的请求地址。<br>不同服务不同区域的endpoint不同，您可以从 <a href="#">终端节点</a> 中获取。 |
| project_id | 是    | 项目ID，您可以从 <a href="#">获取项目ID</a> 中获取。                                   |

## 请求参数

表 4-112 请求 Header 参数

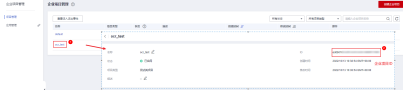
| 参数                    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X-Auth-Token          | 是    | String | 用户Token。<br>用于获取操作API的权限。 <a href="#">获取Token接口</a> 响应消息头中X-Subject-Token的值即为Token。                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Content-Type          | 是    | String | 发送的实体的MIME类型，参数值为“application/json”。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Enterprise-Project-Id | 否    | String | <p>企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。</p> <p>获取方法：进入“<a href="#">企业项目管理</a>”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。</p>  <p>企业项目创建步骤请参见用户指南。</p> <p><b>说明</b><br/>创建企业项目后，在传参时，有以下三类场景。</p> <ul style="list-style-type: none"><li>• 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。</li><li>• 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。</li><li>• 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。</li></ul> |

表 4-113 请求 Body 参数

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| image | 否    | String | 与url二选一。<br>图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。<br>图片最短边不小于15px，最长边不超过8192px，支持JPEG、JPG、PNG、BMP、TIFF格式。<br>图片Base64编码示例如/9j/4AAQSkZJRgABA...，带有多余前缀会产生The image format is not supported报错。                                                                                                                                                                                                       |
| url   | 否    | String | 与image二选一。<br>图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持： <ul style="list-style-type: none"><li>公网http/https url</li><li>OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见<a href="#">配置OBS访问权限</a>。</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li><li>请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li><li>url中不能存在中文字符，若存在，中文需要进行utf8编码。</li></ul> |
| side  | 否    | String | 证件图片正反面信息。可选值包括： <ul style="list-style-type: none"><li>front：越南身份证正面</li><li>back：越南身份证背面</li></ul> 如果参数值为空或无该参数，系统自动识别，建议填写，准确率更高。                                                                                                                                                                                                                                                                                                            |

| 参数                       | 是否必选 | 参数类型    | 描述                                                                                                                                                                             |
|--------------------------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| return_portrait_image    | 否    | Boolean | 是否返回头像内容开关，可选值如下所示： <ul style="list-style-type: none"><li>• true：返回身份证头像照片的 base64编码</li><li>• false：：不返回身份证头像照片的 base64编码</li></ul> 未传入该参数时默认为“false”，即不返回身份证头像照片的 base64 编码。 |
| return_portrait_location | 否    | Boolean | 是否返回头像坐标的开关，可选值如下所示： <ul style="list-style-type: none"><li>• true：返回身份证头像的位置</li><li>• false：不返回身份证头像的位置</li></ul>                                                             |
| return_idcard_type       | 否    | Boolean | 是否返回身份证拍摄类型的开关，可选值如下所示： <ul style="list-style-type: none"><li>• true：返回身份证的类型，类型包括身份证原件以及身份证复印件。</li><li>• false：不返回身份证的类型。</li></ul>                                          |
| return_text_location     | 否    | Boolean | 识别到的文字块的区域位置信息。可选值包括： <ul style="list-style-type: none"><li>• true：返回各个文字块区域</li><li>• false：不返回各个文字块区域</li></ul> 如果无该参数，系统默认不返回文字块区域。如果输入参数不是 Boolean 类型，则会报非法参数错误。           |

响应参数

状态码： 200

表 4-114 响应 Body 参数

| 参数     | 参数类型                             | 描述                    |
|--------|----------------------------------|-----------------------|
| result | <a href="#">表4-115</a><br>object | 调用结果。<br>调用失败时不返回此字段。 |

表 4-115 VietnamIdCardResult

| 参数                      | 参数类型                  | 描述                                                                                                     |
|-------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| side                    | String                | 返回证件正反面。字段值为“front”或“back”                                                                             |
| number                  | String                | 卡证编号。                                                                                                  |
| full_name               | String                | 姓名。                                                                                                    |
| birth_date              | String                | 出生日期。                                                                                                  |
| sex                     | String                | 性别。                                                                                                    |
| nationality             | String                | 国籍。                                                                                                    |
| origin_place            | String                | 籍贯。                                                                                                    |
| residence_place         | String                | 居住地。                                                                                                   |
| expiry_date             | String                | 有效日期。                                                                                                  |
| personal_identification | String                | 个人识别。当参数“side”为back时，返回此字段。                                                                            |
| issue_date              | String                | 签发日期。当参数“side”为back时，返回此字段。                                                                            |
| machine_code_1          | String                | 身份证背面第一行机器码。当参数“side”为back时，返回此字段。                                                                     |
| machine_code_2          | String                | 身份证背面第二行机器码。当参数“side”为back时，返回此字段。                                                                     |
| machine_code_3          | String                | 身份证背面第三行机器码。当参数“side”为back时，返回此字段。                                                                     |
| confidence              | Object                | 相关字段的置信度信息，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。注：置信度由算法给出，不直接等价于对应字段的准确率。                       |
| portrait_image          | String                | 当参数return_portrait_image为true时，返回头像的base64编码。                                                          |
| portrait_location       | Array<Array<Integer>> | 当参数return_portrait_location为true时，返回头像在原图上的位置，以列表形式表示，包含头像区域四个顶点的二维坐标（x,y）；坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 |
| idcard_type             | String                | 输入参数return_idcard_type为true时，返回身份证的类型：normal是身份证原件，copy是复印的身份证，screen是屏幕翻拍。                            |
| text_location           | Object                | 对应所有在原图上识别到的字段位置信息，包含所有文字区域四个顶点的二维坐标（x,y）。采用图像坐标系，坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。                          |

状态码： 400

表 4-116 响应 Body 参数

| 参数         | 参数类型   | 描述                                                    |
|------------|--------|-------------------------------------------------------|
| error_code | String | 调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。<br>调用成功时不返回此字段。 |
| error_msg  | String | 调用失败时返回的错误信息。<br>调用成功时不返回此字段。                         |

## 请求示例

### 说明

- Endpoint即调用API的请求地址，不同服务不同区域的Endpoint不同，具体请参见[终端节点](#)。例如，越南身份证识别服务部署在“中国-香港”区域的“endpoint”为“ocr.ap-southeast-1.myhuaweicloud.com”，请求URL为“https://ocr.ap-southeast-1.myhuaweicloud.com/v2/{project\_id}/ocr/vietnam-id-card”，“project\_id”为项目ID，获取方法请参见[获取项目ID](#)。
- 如何获取Token具体操作请参见[构造请求](#)。

POST https://{endpoint}/v2/{project\_id}/ocr/vietnam-id-card

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

## 响应示例

状态码： 200

成功响应示例

```
{
 "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 Bái",
 "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/4AAQSkZJRgABAQAAQABAA...",
"portrait_location" : [[217, 359], [582, 360], [573, 882], [199, 885]]
}
```

**状态码： 400**

失败响应示例

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

## SDK 代码示例

SDK代码示例如下。

### 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

## Java

- 传入越南身份证图片的base64编码进行文字识别

```
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();
 }
 }
}
```

```
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
```

- 传入越南身份证图片的url进行文字识别

```
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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

- 传入越南身份证图片的base64编码进行文字识别

```
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 = VietnamIdCardBody(
 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)
```

- 传入越南身份证图片的url进行文字识别

```
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 = VietnamIdCardBody(
 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

- 传入越南身份证图片的base64编码进行文字识别

```
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: &imageVietnamIdCardBody,
 }
 response, err := client.RecognizeVietnamIdCard(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 传入越南身份证图片的url进行文字识别

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

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

# 4.15 秘鲁身份证识别

## 功能介绍

识别秘鲁身份证图片中的文字内容，并将识别的结构化结果返回给用户。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-6 秘鲁身份证示例图 1



图 4-7 秘鲁身份证示例图 2



## 约束与限制

- 只支持秘鲁身份证的识别。
- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式的图片。
- 图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 图像中秘鲁身份证区域有效占比超过25%，保证整张秘鲁身份证内容及其边缘包含在图像内。
- 支持图像中秘鲁身份证任意角度的水平旋转。
- 支持少量扭曲，扭曲后图像中的秘鲁身份证长宽比与实际秘鲁身份证相差不超过10%。

- 能处理反光、暗光等干扰的图片但影响识别精度。

## 调用方法

请参见[如何调用API](#)。

## 前提条件

在使用越南身份证识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

POST /v2/{project\_id}/ocr/peru-id-card

表 4-117 路径参数

| 参数         | 是否必选 | 说明                                                                      |
|------------|------|-------------------------------------------------------------------------|
| endpoint   | 是    | 终端节点，即调用API的请求地址。<br>不同服务不同区域的endpoint不同，您可以从 <a href="#">终端节点</a> 中获取。 |
| project_id | 是    | 项目ID，您可以从 <a href="#">获取项目ID</a> 中获取。                                   |

## 请求参数

表 4-118 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                  |
|--------------|------|--------|-------------------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>用于获取操作API的权限。 <a href="#">获取Token接口</a> 响应消息头中X-Subject-Token的值即为Token。 |
| Content-Type | 是    | String | 发送的实体的MIME类型，参数值为“application/json”。                                                |

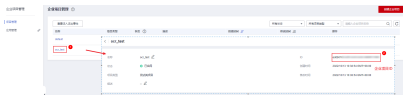
| 参数                    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enterprise-Project-Id | 否    | String | <p>企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。</p> <p>获取方法：进入“<a href="#">企业项目管理</a>”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。</p>  <p>企业项目创建步骤请参见用户指南。</p> <p><b>说明</b></p> <p>创建企业项目后，在传参时，有以下三类场景。</p> <ul style="list-style-type: none"><li>• 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。</li><li>• 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。</li><li>• 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。</li></ul> |

表 4-119 请求 Body 参数

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                      |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| image | 否    | String | <p>与url二选一。</p> <p>图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。</p> <p>图片最小边不小于15像素，最长边不超过8192像素，支持JPG/PNG/BMP/TIFF/PDF格式，支持识别多页PDF数据。</p> <p>图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。</p> |

| 参数                       | 是否必选 | 参数类型    | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------|------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| url                      | 否    | String  | <p>与image二选一。</p> <p>图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持：</p> <ul style="list-style-type: none"><li>• 公网http/https url</li><li>• OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见<a href="#">配置OBS访问权限</a>。</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li><li>• 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li></ul> |
| return_portrait_image    | 否    | Boolean | <p>是否返回头像内容开关，可选值如下所示：</p> <ul style="list-style-type: none"><li>• true: 返回身份证头像照片的base64 编码。</li><li>• false: 返回为空值。未传入该参数时默认为“false”，即返回为空值。</li></ul>                                                                                                                                                                                                                                                                          |
| return_portrait_location | 否    | Boolean | <p>是否返回头像坐标的开关，可选值如下所示：</p> <ul style="list-style-type: none"><li>• true: 返回身份证头像的位置坐标。</li><li>• false: 返回为空值 未传入该参数时默认为“false”，即返回为空。</li></ul>                                                                                                                                                                                                                                                                                 |

响应参数

状态码： 200

表 4-120 响应 Body 参数

| 参数     | 参数类型                                    | 描述                     |
|--------|-----------------------------------------|------------------------|
| result | <a href="#">PeruidCardResult</a> object | 调用成功时表示调用结果。调用失败时无此字段。 |

表 4-121 PeruldCardResult

| 参数                | 参数类型                  | 描述                                                                                                                |
|-------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| cui_number        | String                | 身份证号。                                                                                                             |
| first_surname     | String                | 第一姓氏。                                                                                                             |
| second_surname    | String                | 第二姓氏。                                                                                                             |
| given_name        | String                | 名。                                                                                                                |
| sex               | String                | 性别。                                                                                                               |
| marital_status    | String                | 婚姻状况。                                                                                                             |
| birth_date        | String                | 出生日期。                                                                                                             |
| nationality       | String                | 国籍。                                                                                                               |
| issue_date        | String                | 发行日期。                                                                                                             |
| expiry_date       | String                | 失效日期。                                                                                                             |
| birth_place       | String                | 出生地编码。                                                                                                            |
| voting_group      | String                | 投票组。                                                                                                              |
| organ_donation    | String                | 器官捐赠意愿。                                                                                                           |
| registration_date | String                | 注册日期。                                                                                                             |
| portrait_image    | String                | 头像的base64编码。当输入参数“return_portrait_image”为“true”时，才返回该参数。                                                          |
| portrait_location | Array<Array<Integer>> | 头像在原图上的位置。当输入参数“return_portrait_location”为“true”时，才返回该参数。以列表形式显示，包含头像区域四个顶点的二维坐标（x,y），坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。 |
| address           | String                | 地址。                                                                                                               |
| department        | String                | 大区。                                                                                                               |
| province          | String                | 省。                                                                                                                |
| district          | String                | 区。                                                                                                                |
| remarks           | String                | 备注。                                                                                                               |
| machine_code_1    | String                | 机器码第一行。                                                                                                           |

| 参数             | 参数类型               | 描述                                                                                                      |
|----------------|--------------------|---------------------------------------------------------------------------------------------------------|
| machine_code_2 | String             | 机器码第二行。                                                                                                 |
| machine_code_3 | String             | 机器码第三行。                                                                                                 |
| duplicate      | Boolean            | 是否重新登记过。可选值如下所示： <ul style="list-style-type: none"><li>• true: 已重新登记过</li><li>• false: 未重新登记过</li></ul> |
| confidence     | Map<String,Number> | 相关字段的置信度信息，置信度越大，表示本次识别的对应字段的可靠性越高，在统计意义上，置信度越大，准确率越高。注：置信度由算法给出，不直接等价于对应字段的准确率。                        |

状态码： 400

表 4-122 响应 Body 参数

| 参数         | 参数类型   | 描述                                                    |
|------------|--------|-------------------------------------------------------|
| error_code | String | 调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。<br>调用成功时不返回此字段。 |
| error_msg  | String | 调用失败时返回的错误信息。<br>调用成功时不返回此字段。                         |

## 请求示例

### 说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。  
例如，秘鲁身份证识别服务部署在“拉美-圣地亚哥”区域的“endpoint”为“ocr.la-south-2.myhuaweicloud.com”或“ocr.la-south-2.myhuaweicloud.cn”，请求URL为“https://ocr.la-south-2.myhuaweicloud.com/v2/{project\_id}/peru-id-card”，“project\_id”为项目ID，获取方法请参见[获取项目ID](#)
- 如何获取Token请参见[构造请求](#)。
- 请求示例（方式一：使用图片的Base64编码）  
POST https://{endpoint}/v2/{project\_id}/ocr/peru-id-card  

```
{
 "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...",
 "return_portrait_image": true,
 "return_portrait_location": true
}
```
- 请求示例（方式二：使用图片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
}
```

## 响应示例

状态码: 200

### 成功响应示例

```
{
 "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
 }
 }
}
```

**状态码： 400****失败响应示例**

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

**SDK 代码示例**

SDK代码示例如下。

**📖 说明**

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

**Java**

- 传入秘鲁身份证图片的base64编码进行文字识别

```
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 RecognizePerulCardSolution {

 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();
 RecognizePerulCardRequest request = new RecognizePerulCardRequest();
 PerulCardRequestBody body = new PerulCardRequestBody();
 body.withReturnPortraitLocation(true);
 body.withReturnPortraitImage(true);
 body.withImage("/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA...");
 request.withBody(body);
 try {
 RecognizePerulCardResponse response = client.recognizePerulCard(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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

- 传入秘鲁身份证图片的url进行文字识别

```
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 RecognizePeruldCardSolution {

 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();
 RecognizePeruldCardRequest request = new RecognizePeruldCardRequest();
 PeruldCardRequestBody body = new PeruldCardRequestBody();
 body.withReturnPortraitLocation(true);
 body.withReturnPortraitImage(true);
 body.withUrl("https://BucketName.obs.myhuaweicloud.com/ObjectName");
 request.withBody(body);
 try {
 RecognizePeruldCardResponse response = client.recognizePeruldCard(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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

- 传入秘鲁身份证图片的base64编码进行文字识别

```
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)
```

- 传入秘鲁身份证图片的url进行文字识别

```
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

- 传入秘鲁身份证图片的base64编码进行文字识别

```
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.RecognizePerulCardRequest{
 returnPortraitLocationPerulCardRequestBody:= true
 returnPortraitImagePerulCardRequestBody:= true
 imagePerulCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
 request.Body = &model.PerulCardRequestBody{
 ReturnPortraitLocation: &returnPortraitLocationPerulCardRequestBody,
 ReturnPortraitImage: &returnPortraitImagePerulCardRequestBody,
 Image: &imagePerulCardRequestBody,
 }
 }
 response, err := client.RecognizePerulCard(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 传入秘鲁身份证图片的url进行文字识别

```
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.RecognizePerulCardRequest{
 returnPortraitLocationPerulCardRequestBody:= true
 returnPortraitImagePerulCardRequestBody:= true
 imagePerulCardRequestBody:= "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."
 request.Body = &model.PerulCardRequestBody{
 ReturnPortraitLocation: &returnPortraitLocationPerulCardRequestBody,
 ReturnPortraitImage: &returnPortraitImagePerulCardRequestBody,
 Image: &imagePerulCardRequestBody,
 }
 }
 response, err := client.RecognizePerulCard(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

```
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.RecognizePerulCardRequest{
 returnPortraitLocationPerulCardRequestBody:= true
 returnPortraitImagePerulCardRequestBody:= true
 urlPerulCardRequestBody:= "https://BucketName.obs.myhuaweicloud.com/ObjectName"
 request.Body = &model.PerulCardRequestBody{
 ReturnPortraitLocation: &returnPortraitLocationPerulCardRequestBody,
 ReturnPortraitImage: &returnPortraitImagePerulCardRequestBody,
 Url: &urlPerulCardRequestBody,
 }
 }
 response, err := client.RecognizePerulCard(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.16 泰国车牌识别

### 功能介绍

识别泰国车牌图片中的车牌信息，返回识别结果和车牌的区域位置信息。该接口的使用限制请参见[约束与限制](#)，详细使用指导请参见[OCR服务使用简介](#)章节。

图 4-8 泰国车牌示例图



### 约束与限制

- 只支持识别PNG、JPG、JPEG、BMP、TIFF格式的图片。
- 图像各边的像素大小在15到8192px之间。单个图片对应的Base64编码不超过10MB。
- 支持图像中车牌任意角度的水平旋转。
- 能处理反光、暗光等干扰的图片但影响识别精度。

### 调用方法

请参见[如何调用API](#)。

### 前提条件

在使用泰国车牌识别之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[开通服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

POST /v2/{project\_id}/ocr/thailand-license-plate

表 4-123 路径参数

| 参数         | 是否必选 | 说明                                                                      |
|------------|------|-------------------------------------------------------------------------|
| endpoint   | 是    | 终端节点，即调用API的请求地址。<br>不同服务不同区域的endpoint不同，您可以从 <a href="#">终端节点</a> 中获取。 |
| project_id | 是    | 项目ID，您可以从 <a href="#">获取项目ID</a> 中获取。                                   |

请求参数

表 4-124 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                  |
|--------------|------|--------|-------------------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>用于获取操作API的权限。 <a href="#">获取Token接口</a> 响应消息头中X-Subject-Token的值即为Token。 |
| Content-Type | 是    | String | 发送的实体的MIME类型，参数值为“application/json”。                                                |

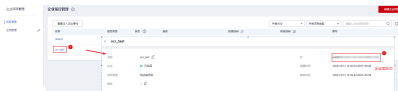
| 参数                    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enterprise-Project-Id | 否    | String | <p>企业项目ID。OCR支持通过企业项目管理（EPS）对不同用户组和用户的资源使用，进行分账。</p> <p>获取方法：进入“<a href="#">企业项目管理</a>”页面，单击企业名称，在企业项目详情页获取Enterprise-Project-Id（企业项目ID）。</p>  <p>企业项目创建步骤请参见用户指南。</p> <p><b>说明</b></p> <p>创建企业项目后，在传参时，有以下三类场景。</p> <ul style="list-style-type: none"><li>• 携带正确的ID，正常使用OCR服务，账单的企业项目会被分类到企业ID对应的企业项目中。</li><li>• 携带格式正确但不存在的ID，正常使用OCR服务，账单的企业项目会显示对应不存在的企业项目ID。</li><li>• 不携带ID或格式错误ID（包含特殊字符等），正常使用OCR服务，账单的企业项目会被分类到"default"中。</li></ul> |

表 4-125 请求 Body 参数

| 参数    | 是否必选 | 参数类型   | 说明                                                                                                                                                                                                                                                     |
|-------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| image | 否    | String | <p>与url二选一。</p> <p>图片的Base64编码，单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。图片尺寸不小于15×15像素，最长边不超过8192像素，支持JPG/PNG/BMP/TIFF/JPEG格式。</p> <p>图片Base64编码示例如/9j/4AAQSkZJRgABAg...，带有多余前缀会产生The image format is not supported报错。</p> |

| 参数  | 是否必选 | 参数类型   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| url | 否    | String | <p>与image二选一。</p> <p>图片的url路径，url中单个图片其对应的Base64编码不超过10MB。图片在Base64编码后会大于图片原本大小，请注意做好边界判断，建议图片大小不超过7MB。目前支持：</p> <ul style="list-style-type: none"><li>• 公网http/https url</li><li>• OBS提供的url，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权，详情参见<a href="#">配置OBS访问权限</a>。</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>• 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li><li>• 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li></ul> |

响应参数

状态码： 200

表 4-126 响应 Body 参数

| 参数     | 参数类型                                                      | 描述                    |
|--------|-----------------------------------------------------------|-----------------------|
| result | Array of <a href="#">ThailandLicensePlateItem</a> objects | 调用结果。<br>调用失败时不返回此字段。 |

表 4-127 ThailandLicensePlateItem

| 参数             | 参数类型                  | 描述                                                                                 |
|----------------|-----------------------|------------------------------------------------------------------------------------|
| plate_number   | String                | 车牌内容。                                                                              |
| plate_location | Array<Array<Integer>> | 车牌的区域位置信息，列表形式，包含文字区域四个顶点的二维坐标（x,y）;坐标原点为图片左上角，x轴沿水平方向，y轴沿竖直方向。                    |
| confidence     | Float                 | 字段的置信度，取值范围0~1。<br>置信度越大，本次识别的字段的可靠性越高，在统计意义上，置信度越大，准确率越高。注：置信度由算法给出，不直接等价于字段的准确率。 |
| province       | String                | 车牌所属府。                                                                             |

**状态码： 400****表 4-128 响应 Body 参数**

| 参数         | 参数类型   | 描述                                                    |
|------------|--------|-------------------------------------------------------|
| error_code | String | 调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。<br>调用成功时不返回此字段。 |
| error_msg  | String | 调用失败时返回的错误信息。<br>调用成功时不返回此字段。                         |

## 请求示例

### 说明

- “endpoint”即调用API的请求地址，不同服务不同区域的“endpoint”不同，具体请参见[终端节点](#)。  
例如，泰国车牌识别服务部署在“亚太-曼谷”区域的“endpoint”为“ocr.ap-southeast-2.myhuaweicloud.com”或“ocr.ap-southeast-2.myhuaweicloud.cn”，请求URL为“https://ocr.ap-southeast-2.myhuaweicloud.com/v2/{project\_id}/ocr/thailand-license-plate”，“project\_id”为项目ID，获取方法请参见[获取项目ID](#)
- 如何获取Token请参见[构造请求](#)。
- 请求样例（方式一：使用图片的Base64编码）  
POST https://{endpoint}/v2/{project\_id}/ocr/thailand-license-plate  
  
Request Header:  
Content-Type: application/json  
X-Auth-Token:  
MIINRwYJKoZIhvcNAQcCoIIODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...  
Request Body:  
{  
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/4RFZRXhpZgAATU0AKgAAAA..."  
}
- 请求样例（方式二：使用图片URL）  
POST https://{endpoint}/v2/{project\_id}/ocr/thailand-license-plate  
  
Request Header:  
Content-Type: application/json  
X-Auth-Token:  
MIINRwYJKoZIhvcNAQcCoIIODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...  
Request Body:  
{  
  "url": "https://BucketName.obs.xxx.com/ObjectName"  
}
- Python3语言请求代码示例（其他语言参照下列示例编写或使用OCR SDK）  
# encoding:utf-8  
  
import requests  
import base64  
  
url = "https://{endpoint}/v2/{project\_id}/ocr/thailand-license-plate"  
token = "用户获取到的实际token值"  
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}  
  
imagepath = r'./data/thailand-license-plate-demo.png' # 读取本地图片  
with open(imagepath, "rb") as bin\_data:  
  image\_data = bin\_data.read()  
  image\_base64 = base64.b64encode(image\_data).decode("utf-8") # 使用图片的Base64编码

```
payload = {"image": image_base64}
response = requests.post(url, headers=headers, json=payload)
print(response.text)
```

## 响应示例

### 状态码：200

#### 成功响应示例

```
{
 "result": [{
 "plate_number": "กข XXX4",
 "province": "มหาสารคาม",
 "confidence": 0.9225,
 "plate_location": [[370, 881], [2591, 881], [2591, 2281], [370, 2281]]
 }]
}
```

### 状态码：400

#### 失败响应示例

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

## SDK 代码示例

SDK代码示例如下。

### 说明

使用SDK前建议将SDK更新至最新版，防止本地旧版SDK无法使用最新的OCR功能。

## Java

- 传入泰国车牌图片的base64编码进行文字识别

```
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.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

- 传入泰国车牌图片的url进行文字识别

```
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

- 传入泰国车牌图片的base64编码进行文字识别

# 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)
```

- 传入泰国车牌图片的url进行文字识别

# 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

- 传入泰国车牌图片的base64编码进行文字识别

```
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)
 }
}
```

- 传入泰国车牌图片的url进行文字识别

```
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)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述     |
|-----|--------|
| 200 | 成功响应示例 |
| 400 | 失败响应示例 |

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

# 5 状态码

HTTP 状态码为三位数，分成五个类别：1xx：相关信息；2xx：操作成功；3xx：重定向；4xx：客户端错误；5xx：服务器错误。

状态码如下所示。

| 状态码 | 编码                            | 状态说明                                                  |
|-----|-------------------------------|-------------------------------------------------------|
| 100 | Continue                      | 继续请求。<br>这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。         |
| 101 | Switching Protocols           | 切换协议。只能切换到更高级的协议。<br>例如，切换到HTTPS的新版本协议。               |
| 200 | OK                            | 服务器已成功处理了请求。                                          |
| 201 | Created                       | 创建类的请求完全成功。                                           |
| 202 | Accepted                      | 已经接受请求，但未处理完成。                                        |
| 203 | Non-Authoritative Information | 非授权信息，请求成功。                                           |
| 204 | No Content                    | 请求完全成功，同时HTTP响应不包含响应体。<br>在响应OPTIONS方法的HTTP请求时返回此状态码。 |
| 205 | Reset Content                 | 重置内容，服务器处理成功。                                         |
| 206 | Partial Content               | 服务器成功处理了部分GET请求。                                      |
| 300 | Multiple Choices              | 多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择。  |
| 301 | Moved Permanently             | 永久移动，请求的资源已被永久的移动到新的URI，返回信息会包括新的URI。                 |
| 302 | Found                         | 资源被临时移动。                                              |

| 状态码 | 编码                            | 状态说明                                                                                    |
|-----|-------------------------------|-----------------------------------------------------------------------------------------|
| 303 | See Other                     | 查看其它地址，使用GET和POST请求查看。                                                                  |
| 304 | Not Modified                  | 所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。                                                          |
| 305 | Use Proxy                     | 所请求的资源必须通过代理访问。                                                                         |
| 306 | Unused                        | 已经被废弃的HTTP状态码。                                                                          |
| 400 | Bad Request                   | 非法请求。<br>建议直接修改该请求，不要重试该请求。                                                             |
| 401 | Unauthorized                  | 在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。                                            |
| 402 | Payment Required              | 保留请求。                                                                                   |
| 403 | Forbidden                     | 请求被拒绝访问。<br>返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多的事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。 |
| 404 | Not Found                     | 所请求的资源不存在。<br>建议直接修改该请求，不要重试该请求。                                                        |
| 405 | Method Not Allowed            | 请求中带有该资源不支持的方法。<br>建议直接修改该请求，不要重试该请求。                                                   |
| 406 | Not Acceptable                | 服务器无法根据客户端请求的内容特性完成请求。                                                                  |
| 407 | Proxy Authentication Required | 请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。                                                      |
| 408 | Request Timeout               | 服务器等候请求时发生超时。<br>客户端可以随时再次提交该请求而无需进行任何更改。                                               |
| 409 | Conflict                      | 服务器在完成请求时发生冲突。<br>返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。                           |
| 410 | Gone                          | 客户端请求的资源已经不存在。<br>返回该状态码，表明请求的资源已被永久删除。                                                 |
| 411 | Length Required               | 服务器无法处理客户端发送的不带Content-Length的请求信息。                                                     |

| 状态码 | 编码                              | 状态说明                                                                                  |
|-----|---------------------------------|---------------------------------------------------------------------------------------|
| 412 | Precondition Failed             | 未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。                                                     |
| 413 | Request Entity Too Large        | 由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。 |
| 414 | Request URI Too Long            | 请求的URI过长（URI通常为网址），服务器无法处理。                                                           |
| 415 | Unsupported Media Type          | 服务器无法处理请求附带的媒体格式。                                                                     |
| 416 | Requested Range Not Satisfiable | 客户端请求的范围无效。                                                                           |
| 417 | Expectation Failed              | 服务器无法满足Expect的请求头信息。                                                                  |
| 422 | Unprocessable Entity            | 请求格式正确，但是由于含有语义错误，无法响应。                                                               |
| 429 | Too Many Requests               | 表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。          |
| 500 | Internal Server Error           | 表明服务端能被请求访问到，但是不能理解用户的请求。                                                             |
| 501 | Not Implemented                 | 服务器不支持请求的功能，无法完成请求。                                                                   |
| 502 | Bad Gateway                     | 充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。                                                        |
| 503 | Service Unavailable             | 被请求的服务无效。<br>建议直接修改该请求，不要重试该请求。                                                       |
| 504 | Gateway Timeout                 | 请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。                                         |
| 505 | HTTP Version Not Supported      | 服务器不支持请求的HTTPS协议的版本，无法完成处理。                                                           |

# 6 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。当调用出错时，HTTPS请求返回一个 4xx 或 5xx 的 HTTPS状态码。返回的消息体中是具体的错误代码及错误信息。在调用方找不到错误原因时，可以联系华为云客服技术支持，并提供错误码，以便尽快帮您解决问题。

## 错误响应 Body 体格式说明

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
 "error_msg": "The input parameter is invalid.",
 "error_code": "AIS.0101"
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

## 错误码说明

由于OCR服务部署在ModelArts平台，使用API网关（API Gateway）系统，因此OCR服务错误码会出现“ModelArts”、“APIGW”、“APIG”开头的错误码。

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。遇到“APIG”开头的错误码，请参考本文档进行处理。

| 模块   | 错误码      | 错误信息                                                                       | 说明                              | 建议解决方法                                                                                                                                                                                            |
|------|----------|----------------------------------------------------------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 文字识别 | AIS.0101 | The input parameter is invalid.                                            | 输入参数不符合规范。                      | <p>请检查输入参数是否正确。出现此报错可能原因如下：</p> <ul style="list-style-type: none"> <li>image、url参数为二选一，请勿同时填写。填写时注意格式规范。</li> <li>功能开关true、false拼写错误。</li> <li>传参时中英文字符错误。</li> </ul> <p>具体参数格式参见各API的请求参数说明。</p> |
|      | AIS.0102 | The image format is not supported.                                         | 图片格式不支持，或image参数值不是base64编码字符串。 | 请检查图片格式，各服务支持图片格式请参考 <a href="#">使用限制</a> 。                                                                                                                                                       |
|      | AIS.0103 | The image size does not meet the requirements.                             | 图片尺寸不满足要求。                      | 请检查图片尺寸，各服务支持图片尺寸请参考 <a href="#">使用限制</a> 。                                                                                                                                                       |
|      | AIS.0104 | The image is not supported or the image quality is poor.                   | 非支持的图片类型或图片质量差。                 | <p>请检查图片类型与图片质量。</p> <p>请检查调用的接口是否正确。</p>                                                                                                                                                         |
|      | AIS.0105 | Recognition failed.                                                        | 算法计算失败。                         | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                                                     |
|      | AIS.0117 | Input ID does not exist.                                                   | 输入ID不存在。                        | 请检查入参中的classifier_id或template_id是否存在。                                                                                                                                                             |
|      | AIS.0118 | Service request error or the tax authority's interface was malfunctioning. | 税局机关接口超时未响应，导致服务请求错误。           | 请等待税局机关接口恢复(建议2小时后重试)。                                                                                                                                                                            |
|      | AIS.0119 | Template match failed, template_id={xxx}.                                  | 输入图片与模板匹配失败。                    | 请检查模板中的参照字段是否与输入图片匹配。                                                                                                                                                                             |
|      | AIS.0120 | Classification failed.                                                     | 输入图片分类失败。                       | 请检查分类器中模板是否与输入图片匹配。                                                                                                                                                                               |

| 模块 | 错误码            | 错误信息                                                                  | 说明               | 建议解决方法                                                            |
|----|----------------|-----------------------------------------------------------------------|------------------|-------------------------------------------------------------------|
|    | AIS.0121       | The recognition area of template is not configured, template_id={xxx} | 模板未配置识别区。        | 请检查模板是否配置了识别区。                                                    |
|    | AIS.0122       | The image contains two or more ID cards with the same side.           | 存在两张及以上同面身份证。    | 请检查输入图片是否存在两张及以上同面身份证。身份证服务参数side为double_side时，图片中不能存在两张及以上同面身份证。 |
| 平台 | ModelArts.0203 | Invalid token.                                                        | 非法Token。         | 请检查Token是否正确。                                                     |
|    | ModelArts.4101 | Token header cannot be empty.                                         | Token为空。         | 发送的HTTP请求header中没有带x-auth-token的token请求认证信息，请检查发送的请求。             |
|    | ModelArts.4102 | Parse Token error.                                                    | 解析Token失败。       | 发送的HTTP请求header中x-auth-token的token请求认证信息错误，请检查发送的请求和token。        |
|    | ModelArts.4103 | Invalid Token header.                                                 | Token不合法。        | 发送的HTTP请求header中x-auth-token的token请求认证信息错误，请检查发送的请求和token。        |
|    | ModelArts.4104 | Invalid Request Content Length.                                       | 请求body长度不合法。     | 请检查请求body长度。要求Base64编码后大小不超过10MB。                                 |
|    | ModelArts.4105 | The JSON format of the input data is incorrect.                       | 请求body JSON格式不对。 | 请检查请求body是否符合JSON格式。                                              |
|    |                | Failed to obtain the temporary AK/SK                                  | 获取用户临时AK，SK失败    | 请联系技术支持人员（如果有使用OBS，请先检查授权）。                                       |
|    | ModelArts.4106 | Invalid authorization request.                                        | 用户账号受限。          | 请检查用户资源状态。账号受限原因请参考 <a href="#">帐号中心相关描述</a> 。                    |

| 模块 | 错误码            | 错误信息                                              | 说明                    | 建议解决方法                                                                                                                                                                   |
|----|----------------|---------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | ModelArts.4107 | Get user temp ak sk error.                        | 获取用户临时AK，SK异常。        | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4201 | Request url need service id.                      | 请求url中需要包含服务ID。       | 请检查请求url中的服务ID。                                                                                                                                                          |
|    | ModelArts.4202 | Request url format invalid.                       | 请求url格式不合法。           | 请检查请求url格式。                                                                                                                                                              |
|    | ModelArts.4203 | Access denied! You do not have permission.        | 没有权限访问。               | 请检查访问权限。                                                                                                                                                                 |
|    | ModelArts.4204 | Request api error! The API XXX is not subscribed. | 请求服务失败，服务未开通。         | <ul style="list-style-type: none"> <li>请前往<a href="#">文字识别</a>控制台开通需要调用的API，注意API的开通区域。</li> <li>如已开通服务，请检查开通服务的区域（或账号）与调用服务的区域（或账号）是否一致；若一致请检查API的URL是否拼写正确。</li> </ul> |
|    | ModelArts.4301 | Have error when get ai-service admin token.       | 获取服务admin Token失败。    | 请联系华为云支持人员。                                                                                                                                                              |
|    | ModelArts.4302 | Gateway forwarding error.                         | 访问服务失败。               | 请联系华为云支持人员。                                                                                                                                                              |
|    | ModelArts.4401 | Max concurrency error.                            | 最大并发错误，请求并发过大，超过限额。   | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4402 | Backend service timeout error.                    | 服务处理超时。               | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4403 | Backend service unavailable error.                | 服务无法正常响应，或者失败率过高，不可用。 | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4502 | Backend service response error.                   | 服务响应失败。               | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4503 | Backend service found error.                      | 服务不存在。                | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |
|    | ModelArts.4504 | Backend service api not found.                    | 访问的API不存在。            | 请 <a href="#">联系华为云支持人员</a> 。                                                                                                                                            |

| 模块 | 错误码            | 错误信息                                                                                    | 说明             | 建议解决方法                         |
|----|----------------|-----------------------------------------------------------------------------------------|----------------|--------------------------------|
|    | ModelArts.4505 | Backend service internal error.                                                         | 服务内部错误。        | 请 <a href="#">联系华为云支持人员</a> 。  |
|    | ModelArts.4506 | SubService service found error.                                                         | 该API对应的子服务不存在。 | 请 <a href="#">联系华为云支持人员</a> 。  |
|    | ModelArts.4508 | Service not start.                                                                      | 服务不存在或者服务停止。   | 请 <a href="#">联系华为云支持人员</a> 。  |
|    | ModelArts.4601 | Failed to download the file because the file path is not valid, please check it format. | 外网下载地址不合法。     | 请检查输入的url下载地址参数格式。             |
|    | ModelArts.4603 | Obtaining the file from the URL failed.                                                 | 下载外网URL文件失败。   | 请检查网络与URL。                     |
|    | ModelArts.4702 | Query Obs agency failed.                                                                | 查询OBS委托失败。     | 请检查是否已开通服务的OBS委托。              |
|    | ModelArts.4703 | The Obs URL is invalid.                                                                 | OBS地址不合法。      | 请检查OBS地址。                      |
|    | ModelArts.4704 | Obtaining the file from the OBS failed.                                                 | 获取OBS文件失败。     | 请检查OBS文件。                      |
|    | ModelArts.4705 | The file stored on the OBS is oversized.                                                | OBS文件大小超限。     | 请检查OBS文件大小，请使用符合服务大小限制的文件作为输入。 |
|    | ModelArts.4706 | The Obs file is not exist.                                                              | OBS文件不存在。      | 请检查对应的文件是否存在。                  |
|    | ModelArts.6201 | The user account has been suspended. Please check if you have an outstanding balance.   | 用户账户被冻结。       | 请检查账户是否被冻结。请检查账户余额并完成充值。       |

| 模块   | 错误码       | 错误信息                                                                 | 说明                  | 建议解决方法                                                                                                                                                                                                                                                 |
|------|-----------|----------------------------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| APIG | APIG.0101 | The API does not exist or has not been published in the environment. | 访问的API不存在或尚未在环境中发布。 | <ul style="list-style-type: none"> <li>请检查API的URL是否拼写正确，例如，URL中是否缺少project_id。</li> <li>检查URI中的区域信息是否和调用服务配置的终端节点是否对应。</li> <li>HTTP请求方法（POST，GET等）是否正确。</li> <li>具体的URI信息请在各接口的API页面查看。</li> </ul>                                                    |
|      | APIG.0201 | Backend timeout.                                                     | 请求超时。               | <ul style="list-style-type: none"> <li>请检查原调用请求是否过于频繁，如果是并发过大，可以通过重试机制解决，在代码里检查返回值，碰到这个并发错误可以延时一小段时间（如2-5s）重试请求；也可以后端检查上一个请求结果，上一个请求返回之后再发送下一个请求，避免请求过于频繁。</li> <li>请检查图片是否过大，或网络延时过长，如果图片过大，在保证图片清晰度前提下可以对图片进行等比例压缩；如果网络延时较长，可以适当提升网络传输速度。</li> </ul> |

| 模块 | 错误码                                          | 错误信息                                                                                          | 说明                                                                                                                                                                                                                                         | 建议解决方法                                                                                                                                                                                                                                                                                                                                                                        |
|----|----------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | APIG.0301                                    | Incorrect IAM authentication information.                                                     | <p>IAM身份验证信息不正确：</p> <ul style="list-style-type: none"> <li>decrypt token fail: token解析失败。</li> <li>token expires: token过期。</li> <li>verify aksk signature fail: AK/SK认证失败。</li> <li>x-auth-token not found: 未找到x-auth-token参数。</li> </ul> | <ul style="list-style-type: none"> <li>token解析失败，请检查获取token的方法，请求体信息是否填写正确，token是否正确；检查获取token的环境与调用的环境是否一致。</li> <li>token超时（token expires），请重新获取token，使用不过期的token。</li> <li>请检查AK/SK是否正确（AK对应的SK错误，不匹配；AK/SK中多填了空格）。</li> <li>AK/SK频繁出现鉴权出错，连续错误5次以上，被锁定5分钟（5分钟内，则一直认为其是异常的鉴权请求），5分钟后解锁重新认证。</li> <li>检查账号权限，是否欠费，被冻结等。</li> <li>检查请求header参数X-Auth-Token是否拼写正确。</li> </ul> |
|    | APIG.0308                                    | The throttling threshold has been reached: policy user over ratelimit,limit:XX,time:1 minute. | 发送请求超过了服务的默认配置限流。                                                                                                                                                                                                                          | 请参考 <a href="#">调用API时提示"APIG.0308"</a> 方案进行解决。                                                                                                                                                                                                                                                                                                                               |
| 其他 | 如若出现其他无法解决的错误码，请 <a href="#">联系华为云支持人员</a> 。 |                                                                                               |                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                               |

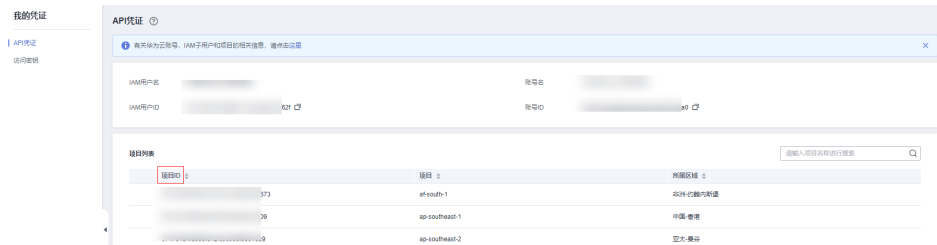
# 7 附录

## 7.1 获取项目 ID

### 从控制台获取项目 ID

1. 登录[管理控制台](#)。
2. 鼠标移动到右上角的用户名上，在下拉列表中选择“我的凭证”。
3. 在“我的凭证 > API凭证”页面，可以查看用户名、账号名，在项目列表中查看项目。

图 7-1 查看项目 ID



多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。

### 调用 API 获取项目 ID

获取项目ID的接口为“GET https://{endpoint}/v3/projects”，其中“{endpoint}”为IAM的终端节点。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，例如，文字识别服务部署的区域为“ap-southeast-1”，响应消息体中查找“name”为“ap-southeast-1”，其中“projects”下的“id”即为项目ID。获取项目ID报错时，请检查{endpoint}是否填写正确。

```
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 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id），所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 注册并登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。  
在“我的凭证”页面的项目列表中查看账号ID。

图 7-2 查看账号 ID



## 7.3 配置 OBS 访问权限

EI企业智能服务对于图片、语音等多媒体文件支持直接使用OBS服务的数据处理方式，以减少服务使用成本，降低服务的响应时长，提升服务使用的体验。

考虑到数据的安全，当对服务进行授权后，才可以使用授权方式的URL（<https://<bucket-name>.<endpoint>/<object-name>>）对服务进行访问。若未对服务授权，则无法直接获取到用户数据，需要用户开启公共读授权或者提供一个临时授权的URL。

### 对文字识别服务开启授权

如果您需要使用OBS中的数据，请开通对象存储服务OBS授权。进入文字识别[控制台](#)。打开对象存储服务OBS授权的按钮，完成授权操作。完成授权即可使用授权方式的URL对服务进行访问。

图 7-3 OBS 授权

如果您想使用存储在对象存储服务（OBS）上的数据，您可以 [服务授权](#) 文字识别服务访问您的对象存储服务。

### 说明

不支持跨区域OBS，OBS的区域需要和服务保持一致。

OBS特性的使用限制请参考[约束与限制](#)。

## 开启公共读授权（可选）

配置公共读可参考《对象存储服务控制台指南》[配置标准桶策略](#)，创建桶策略为“公共读”。一般私密数据不建议用此方法。

## 使用临时授权请求鉴权

开启公共读授权访问，虽然使用比较方便，但若对于敏感的信息，例如个人的私有数据，存在泄露风险。此场景下，可以考虑OBS提供的临时授权功能。

OBS服务支持用户对OBS服务中的对象构造一个特定URL，URL中会包含鉴权信息，任何用户都可以通过该URL访问OBS中的对象，但该URL只在Expires指定的失效时间内有效。该方式用于在不提供给其他人Secret Access Key的情况下，让其他人能够执行自己定义的操作。

进一步了解和使用的OBS临时授权功能，请参见《[对象存储服务SDK参考](#)》对应语言的“授权访问”章节的相关内容，下载相关的SDK及示例代码，并进行相关的编码开发，以支持相关的URL获取。