

文字识别

SDK 参考

文档版本

01

发布日期

2020-11-27



版权所有 © 华为技术有限公司 2020。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，华为公司对本文档内容不做任何明示或默示的声明或保证。

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。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目录

1 文字识别 SDK 简介	1
2 文字识别服务 SDK 环境搭建	3
2.1 Java 开发环境配置.....	3
2.2 Python 开发环境配置.....	5
2.3 iOS 开发环境配置.....	6
2.4 Android 开发环境配置.....	6
2.5 Node.js 开发环境配置.....	7
3 使用 SDK (Java)	9
4 使用 SDK (Python)	11
5 使用 SDK (iOS)	13
6 使用 SDK (Android)	15
7 使用 SDK (Node.js)	17
A 修订记录	19

1 文字识别 SDK 简介

文字识别概述

文字识别（Optical Character Recognition，简称OCR）将图片或扫描件中的文字识别成可编辑的文本。可代替人工录入，提升业务效率。支持身份证、驾驶证、行驶证、发票、通用表格、通用文字等场景文字识别。

文字识别以开放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，帮助用户提升业务效率。

SDK 概述

文字识别软件开发工具包（Optical Character Recognition Software Development Kit，简称OCR SDK）是对文字识别提供的REST API进行的封装，以简化用户的开发工作。用户直接调用OCR SDK提供的接口函数即可实现使用文字识别服务业务能力的目的。

接口与 API 对应关系

文字识别接口与API对应关系请参见[表1-1](#)。

表 1-1 接口与 API 对应关系表

接口	API
通用表格识别	POST /v1.0/ocr/general-table
通用文字识别	POST /v1.0/ocr/general-text
网络图片识别	POST /v1.0/ocr/web-image
智能分类识别	POST /v1.0/ocr/auto-classification
手写文字识别	POST /v1.0/ocr/handwriting
身份证识别	POST /v1.0/ocr/id-card
行驶证识别	POST /v1.0/ocr/vehicle-license
驾驶证识别	POST /v1.0/ocr/driver-license

接口	API
护照识别	POST /v1.0/ocr/passport
银行卡识别	POST /v1.0/ocr/bankcard
营业执照识别	POST /v1.0/ocr/business-license
道路运输证识别	POST /v1.0/ocr/transportation-license
车牌识别	POST /v1.0/ocr/license-plate
增值税发票识别	POST /v1.0/ocr/vat-invoice
机动车销售发票识别	POST /v1.0/ocr/mvs-invoice
出租车发票识别	POST /v1.0/ocr/taxi-invoice
火车票识别	POST /v1.0/ocr/train-ticket
定额发票识别	POST /v1.0/ocr/quota-invoice
车辆通行费发票识别	POST /v1.0/ocr/toll-invoice
飞机行程单识别	POST /v1.0/ocr/flight-itinerary

2 文字识别服务 SDK 环境搭建

2.1 Java 开发环境配置

操作场景

文字识别服务JAVA SDK支持Windows、Linux、Mac等操作系统。本节以Windows系统为例进行环境配置，要求的操作环境请参见[表2-1](#)。

表 2-1 准备环境

准备环境	说明
操作系统	Windows系统，推荐Windows 7及以上版本。
安装JDK	开发环境的基本配置。JDK版本必须不低于JDK8。
安装和配置Eclipse	用于开发OCR应用程序的工具。

操作步骤

以下步骤以win7环境配置JDK8 64位为例，已经下载JDK并配置好环境请跳过步骤1-4。以eclipse工具为例给出文字识别服务SDK工程导入步骤，如果使用其他IDE工具，请参照处理。

1. [下载JDK文件](#)。
2. 下载完成后按照提示安装，位置自选，例如安装到本地C:\Program Files\Java\jdk1.8.0_131。
3. 配置Java环境变量：右键“计算机>属性>高级系统设置>环境变量”，进行如下操作。
 - a. 新建系统变量JAVA_HOME，变量值为实际JDK安装位置。
 - b. 在Path中添加%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin（注意用英文分号分隔）。

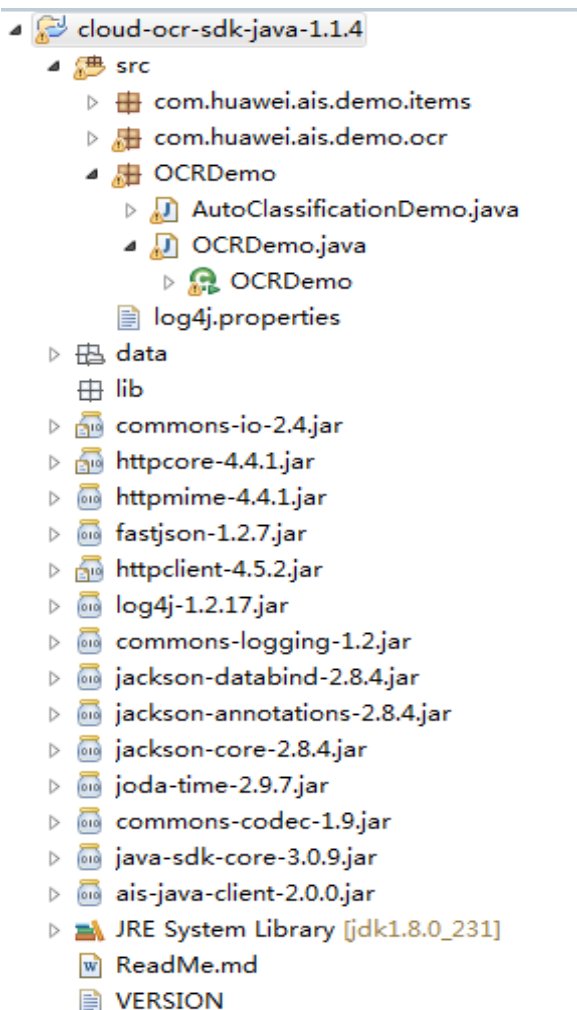
4. 打开命令行窗口，输入“java -version”，显示如图2-1表示配置成功。

图 2-1 Java 版本信息

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

5. Eclipse安装。
在<https://www.eclipse.org/downloads/packages/>下载对应平台的eclipse版本，比如：eclipse-jee-mars-R-win32-x86_64.zip。解压后直接打开eclipse。确保“Window>Preferences>Java>Installed JREs”配置正确的JRE路径。
6. JAVA SDK工程导入。
 - a. 在左侧“Package Explorer”页面右键，单击“Import”。
 - b. 选择“General>Existing Projects into Workspace”。
 - c. 单击“Next”，选择Maven依赖模式、选择文字识别服务SDK Demo包所在路径。
 - d. 单击“Finish”，导入demo，导入后打开工程，工程目录如下。

图 2-2 工程目录



2.2 Python 开发环境配置

操作场景

文字识别服务Python SDK支持Windows、Linux、Mac等操作系统。本节以Windows系统为例进行环境配置，要求的操作环境请参见表2-2。

表 2-2 准备环境

准备环境	说明
操作系统	Windows系统，推荐Windows 7及以上版本。
安装Python	Python版本建议使用2.7及以上版本。
安装Python依赖库	OCR Python SDK依赖第三方库requests 2.18.2包。

操作步骤

以下步骤以win7环境配置Python 2.7为例，已配置好Python2.7及更高版本并且安装了requests 2.18.2包请跳过本章节。

1. 根据系统环境下载Python安装包：<https://www.python.org/downloads/release/python-2711/>。
2. 下载完成后按照提示安装，位置自选，默认位置为C:\Python27。
3. 配置Python环境变量: 右键“计算机>属性>高级系统设置>环境变量”，在path中添加C:\Python27;C:\Python27\Scripts。
4. 打开命令行窗口，输入“python -v”，显示如图2-3表示配置成功。

图 2-3 Python 版本信息

```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec 5 2015, 20:40:30)
AMD64) on win32
Type "help", "copyright", "credits" or "license" for more in
>>>
```

5. 通过PIP在线安装依赖包requests，在新打开的CMD窗口执行如下命令。
`pip install -i https://pypi.tuna.tsinghua.edu.cn/simple requests --trusted-host https://pypi.tuna.tsinghua.edu.cn/simple`
6. 安装完成后通过PIP命令确认requests正确安装，如图2-4所示。

图 2-4 requests 安装确认

```
C:\Users\g00190574>pip list
Package      Version
-----
certifi      2017.4.17
chardet      3.0.4
idna         2.5
pip          10.0.1
requests     2.19.1
setuptools   18.2
urllib3      1.21.1
```

2.3 iOS 开发环境配置

操作场景

文字识别服务iOS SDK包要求的操作环境请参见[表2-3](#)。

表 2-3 准备环境

准备环境	说明
操作系统	iOS 9.0及以上版本，必须在macOS运行。
安装Xcode	Xcode版本建议使用10.0及以上版本。

操作步骤

Xcode是iOS开发的IDE必不可少，以下为步骤以安装Xcode 10.0为例创建iOS开发环境，已配置并安装成功请跳过本章节。

1. 注册Apple ID。
2. 在App Store搜索Xcode，单击“获取”下载。
3. 下载完成之后单击Lanchpad，可以看到Xcode，打开即可。

2.4 Android 开发环境配置

操作场景

文字识别服务Android SDK支持Windows、Linux、Mac等操作系统。本节以Windows系统为例进行环境配置，要求的操作环境请参见[表2-4](#)。

表 2-4 准备环境

准备环境	说明
操作系统	Windows系统，推荐Windows 7及以上版本。
安装Android Studio	Android Studio版本建议使用3.1.3及以上版本。

操作步骤

文字识别Android SDK使用Android Studio IDE开发。以下步骤以win7环境配置Android Studio3.1.3正式版为例，已配置并安装成功请跳过本章节。

1. Java开发环境配置请参见[Java开发环境配置](#)。
2. 配置Android Studio。
访问<https://www.androiddevtools.cn/>，下载3.1.3正式版Windows Android Studio安装包。
3. 启动Android Studio，选择Open an existing Android Studio project，然后选择文字识别Android SDK包的本地路径。
4. 工程的目标API是26，如果本地SDK没有此版本，请安装Android API 26。

2.5 Node.js 开发环境配置

操作场景

文字识别服务Node.js SDK支持Windows、Linux、Mac等操作系统。本节以Windows系统为例进行环境配置，要求的操作环境请参见[表2-5](#)。

表 2-5 准备环境

准备环境	说明
操作系统	Windows系统，推荐Windows 7及以上版本。
安装Node.js	Node.js版本建议使用Node.js v10.13.0及以上版本。
安装Node.js依赖库	OCR Node.js SDK依赖第三方库request、moment、moment-timezone包。

操作步骤

以下步骤以win7环境安装Node.js v10.13.0为例，已安装Node.js或者更高版本并且安装了依赖包请跳过本章节。

1. 配置Node.js。
Node.js下载地址：<http://nodejs.cn/download/>，下载Node.js最新版本。
配置Node.js环境变量: 右键“计算机>属性>高级系统设置>环境变量”在Path中添加C:\Program Files\nodejs (node.exe默认安装路径)。

2. 验证Node.js。

在系统命令行输入`node -v`查看Node.js的版本，如果出现了当前的版本号，即为配置成功。

3. 配置cnpm与配置国内软件库。

```
$ npm install -g cnpm --registry=https://registry.npm.taobao.org
```

如果已经配置，请直接跳转到4。

4. 安装依赖包。

在Node.js项目的根目录中，shift+鼠标右键，选择“在此处打开命令窗口”，执行下列命令。

```
cnpm install request --save  
cnpm install moment --save  
cnpm install moment-timezone --save
```

3 使用 SDK (Java)

用户使用服务的认证方式有Token和AK/SK两种，获取认证消息请参考[认证鉴权](#)。本章节主要包含Token和AK/SK两种方式使用SDK进行示例说明。

在OCR SDK开发工具包地址：<https://developer.huaweicloud.com/sdk?OCR>，选择OCR Java SDK工具包下载并解压。

说明

- SDK中涉及到的所有图像均为合成的非真实图像，仅供示例参考使用。
- 用户首次使用需要先[申请开通](#)该服务。（服务只需要开通一次即可，后面使用时无需再次申请。）

AK/SK 认证方式

本节以身份证识别服务为例介绍如何以AK/SK认证方式使用SDK。

1. 获取AK/SK，具体步骤请参见[认证鉴权>AK/SK](#)。
2. 配置JAVA SDK的AK/SK。

根据获取的AK/SK，修改Demo工程“OCRDemo.java”文件中AKSKDemo函数的“AK”和“SK”的值。

图 3-1 修改 OCRDemo.java 文件参数

```
public static void AKSKDemo() throws URISyntaxException, Un:  
    /*  
     * AK/SK demo code  
     */  
    String AK="xxx"; //AK from authentication  
    String SK="xxx"; //SK from authentication  
    String regionName="cn-north-4"; //region name of the ser  
    String httpUri = "/v1.0/ocr/id-card";  
    String imgPath = "./data/id-card-demo.jpg"; //File path
```

3. 修改输入参数。
如使用本地图片文件进行识别，修改OCRDemo.java文件参数"imgPath"为本地图片路径。如使用SDK默认图片则不需要进行修改。
4. 执行OCRDemo.java文件，控制台输出200即表示程序执行成功。在控制台可以查看身份证识别的结果。

```
{
  "result": {
    "name": "xx",
    "sex": "女",
    "ethnicity": "满",
    "birth": "1990-xx-xx",
    "address": "河北省承德市围场满族蒙古族自治县金车路",
    "number": "3892011990012xxxxx"
  }
}
```

📖 说明

如果想调用OCR其他识别服务，只需要把AKSKDemo函数的"httpUri"参数改成对应的URI即可。URI列表请参见[表1 接口与API对应关系表](#)。

Token 认证方式

本节以身份证识别服务为例介绍如何以Token认证方式使用SDK。

1. 打开OCRDemo包下面的OCRDemo.java文件，修改TokenDemo函数中的username、password和domainName为系统中实际注册的用户名、密码和账号名（如果用户为非IAM用户，账号名与用户名一致）。

图 3-2 修改 OCRDemo.java 文件中的用户名、密码和账号名

```
public static void TokenDemo() throws URISyntaxException, Ur
    /*
     * Token demo code
     */
    String username="xxx";
    String password="xxx";
    String domainName="xxx"; //If the current user is not ar
    String regionName="cn-north-4";
    String httpUri = "/v1.0/ocr/id-card";
    String imgPath = "./data/id-card-demo.jpg"; //File path
```

2. 执行相应代码，在控制台可看到使用Token方式身份证识别服务的识别结果。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

4 使用 SDK (Python)

用户使用服务的认证方式有Token和AK/SK两种，获取认证消息请参考[认证鉴权](#)。本章节主要包含Token和AK/SK两种方式使用SDK进行示例说明。

在OCR SDK开发工具包地址：<https://developer.huaweicloud.com/sdk?OCR>，选择OCR Python SDK工具包下载并解压。

📖 说明

用户首次使用需要先[申请开通](#)该服务。（服务只需要开通一次即可，后面使用时无需再次申请。）

AK/SK 认证方式

本节以身份证识别服务为例介绍如何以AK/SK认证方式使用SDK。

1. 获取AK/SK，具体步骤请参见[认证鉴权>AK/SK认证](#)。
2. 配置Python SDK的AK/SK。

根据获取的AK/SK，修改Demo工程“OCRDemo.py”文件中aksk_request函数的“AK”和“SK”的值。

图 4-1 修改 OCRDemo.py 文件参数

```
def aksk_request():  
    """  
    AK SK demo code  
    """  
    AK = 'xxx' # AK from authentication.  
    SK = 'xxx' # SK from authentication.  
    region = 'cn-north-4' # http region information.  
    req_uri = '/v1.0/ocr/id-card'  
    img_path = './data/id-card-demo.jpg' # File path or URL of the image to be recognized.
```

3. 修改输入参数。

如使用本地图片文件进行识别，修改OCRDemo.py文件参数“img_path”为本地图片路径。如使用SDK默认图片则不需要进行修改。

4. 执行OCRDemo.py文件，控制台输出200即表示程序执行成功。在控制台可以查看身份证的结果。

```
{  
  "result": {
```

```
"name": "xx",  
"sex": "女",  
"ethnicity": "满",  
"birth": "1990-xx-xx",  
"address": "河北省承德市围场满族蒙古族自治县金车路",  
"number": "3892011990012xxxxx"  
}  
}
```

📖 说明

如想调用OCR其他识别服务，只需要把aksk_request函数的"req_uri"参数改成对应的URI即可。URI列表请参见[表1 接口与API对应关系表](#)。

Token 认证方式

本节以身份证识别服务为例介绍如何以Token认证方式使用SDK。

1. 打开OCRDemo.py文件，修改token_request函数中的username、password和domain_name为系统中实际注册的用户名、密码和账号名（如果用户为非IAM用户，账号名与用户名一致）。

图 4-2 修改 OCRDemo.py 文件中的用户名、密码和账号名

```
def token_request():  
    """  
    Token demo code  
    """  
    username = "xxx"  
    password = "xxx"  
    domain_name = "xxx" # If the current us  
    region = "cn-north-4" # cn-north-1, cn-  
    req_uri = "/v1.0/ocr/id-card"  
    img_path = "./data/id-card-demo.jpg"
```

2. 直接执行相应代码，在控制台可看到使用Token方式身份证识别服务的识别结果。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5 使用 SDK (iOS)

用户使用服务的认证方式有Token和AK/SK两种，获取认证消息请参考[认证鉴权](#)。本章节主要包含Token和AK/SK两种方式使用SDK进行示例说明。

在OCR SDK开发工具包地址：<https://developer.huaweicloud.com/sdk?OCR>，选择OCR iOS SDK工具包下载并解压。

📖 说明

用户首次使用需要先[申请开通](#)该服务。（服务只需要开通一次即可，后面使用时无需再次申请。）

AK/SK 认证方式

本节以身份证识别服务为例介绍如何以AK/SK认证方式使用SDK。

1. 获取AK/SK，具体步骤请参见[认证鉴权>AK/SK认证](#)。
2. 配置iOS SDK的AK/SK。

根据获取的AK/SK，修改Demo工程“ViewController.m”文件中akskOcrService函数的“AK”和“SK”的值。

图 5-1 修改 ViewController.m 文件参数

```
- (void)akskOcrService{
# warning Set required parameters
    NSString *AK = @"xxx";// AK from authentication
    NSString *SK = @"xxx";// SK from authentication
    NSString *region = @"cn-north-4";// http endpoint information
    NSString *uri = @"/v1.0/ocr/id-card";
    UIImage *image = [UIImage imageNamed:@"id-card-demo"];
```

3. 如使用本地图片文件进行识别，修改ViewController.m文件中akskOcrService函数中image参数。如使用SDK默认图片则不需要进行修改。
4. 执行akskOcrService函数，控制台输出200即表示程序执行成功。在控制台可以查看身份证识别的结果。

```
{
    "result": {
        "name": "xx",
        "sex": "女",
        "ethnicity": "满",
        "birth": "1990-xx-xx",
```



```
"address": "河北省承德市围场满族蒙古族自治县金车路",  
"number": "3892011990012xxxxx"  
}  
}
```

📖 说明

如想调用OCR其他识别服务，只需要把aksOcrService函数的“uri”参数改成对应的URI即可。URI列表请参见[表1 接口与API对应关系表](#)。

Token 认证方式

本节以身份证识别服务为例介绍如何以Token认证方式使用SDK。

1. 打开ViewController.m文件，修改tokenOcrService函数中的username、password和domainName为系统中实际注册的用户名、密码和账号名（如果用户为非IAM用户，账号名与用户名一致）。

图 5-2 修改 ViewController.m 文件中的用户名、密码和账号名

```
- (void)tokenOcrService{  
# warning Set required parameters  
    NSString *domainName = @"xxx";// if the user isn't IAM user.  
    NSString *userName = @"xxx";  
    NSString *password = @"xxx";  
    NSString *region = @"cn-north-4";  
    NSString *uri = @"/v1.0/ocr/id-card";  
    UIImage *image = [UIImage imageNamed:@"id-card-demo"];
```

2. 直接执行相应代码，在控制台可看到使用Token方式身份证识别服务的识别结果。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

6 使用 SDK (Android)

用户使用服务的认证方式有Token和AK/SK两种，获取认证消息请参考[认证鉴权](#)。本章节主要包含Token和AK/SK两种方式使用SDK进行示例说明。

在OCR SDK开发工具包地址：<https://developer.huaweicloud.com/sdk?OCR>，选择OCR Android SDK工具包下载并解压。

📖 说明

用户首次使用需要先[申请开通](#)该服务。（服务只需要开通一次即可，后面使用时无需再次申请。）

AK/SK 认证方式

本节以身份证识别服务为例介绍如何以AK/SK认证方式使用SDK。

1. 获取AK/SK，具体步骤请参见[认证鉴权>AK/SK认证](#)。
2. 配置Android SDK的AK/SK。

根据获取的AK/SK，修改Demo工程“MainActivity.java”文件中akskOcrService函数的“AK”和“SK”的值。

图 6-1 修改 MainActivity.java 文件参数

```
public void akskOcrService() {  
    // TODO: Set required parameters  
    String AK = "xxx";// AK from authentication  
    String SK = "xxx";// SK from authentication  
    String region = "cn-north-4";  
    // initialize HWOcrClientAKSK from ak,sk and endpoint information  
    HWOcrClientAKSK hwOcrClientAKSK = new HWOcrClientAKSK(this, AK, SK, region);  
    // ocr service  
    String uri = "/v1.0/ocr/id-card";  
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.idcarddemo);  
}
```

3. 修改输入参数。
如使用本地图片文件进行识别，修改MainActivity.java文件中akskOcrService函数的Bitmap类型的参数。如使用SDK默认图片则不需要进行修改。
4. 执行akskOcrService函数，控制台输出200即表示程序执行成功。在控制台可以查看身份证识别的结果。

```
{  
  "result": {
```

```
"name": "xx",  
"sex": "女",  
"ethnicity": "满",  
"birth": "1990-xx-xx",  
"address": "河北省承德市围场满族蒙古族自治县金车路",  
"number": "3892011990012xxxxx"  
}  
}
```

📖 说明

如想调用OCR其他识别服务，只需要把aksOcrService函数的“uri”参数改成对应的URI即可。URI列表请参见[表1 接口与API对应关系表](#)。

Token 认证方式

本节以身份证识别服务为例介绍如何以Token认证方式使用SDK。

1. 打开MainActivity.java文件，修改tokenOcrService函数中的userName、password和domainName为系统中实际注册的用户名、密码和账号名（如果用户为非IAM用户，账号名与用户名一致）。

图 6-2 修改 MainActivity.java 文件中的用户名、密码和账号名

```
public void tokenOcrService() {  
    // TODO: Set required parameters  
    String domainName = "xxx"; // if the user isn't IAM user, domain_name is the same with username  
    String userName = "xxx";  
    String password = "xxx";  
    String region = "cn-north-4";  
    // get ocr service token  
    HWOcrClientToken ocrToken = new HWOcrClientToken(this, domainName, userName, password, region);  
    // ocr service  
    String uri = "/v1.0/ocr/id-card";  
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.idcarddemo);
```

2. 直接执行相应代码，在控制台可看到使用Token方式身份证识别服务的识别结果。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

7 使用 SDK (Node.js)

用户使用服务的认证方式有Token和AK/SK两种，获取认证消息请参考[认证鉴权](#)。本章节主要包含Token和AK/SK两种方式使用SDK进行示例说明。

在OCR SDK开发工具包地址：<https://developer.huaweicloud.com/sdk?OCR>，选择OCR Node.js SDK工具包下载并解压。

📖 说明

用户首次使用需要先[申请开通](#)该服务。（服务只需要开通一次即可，后面使用时无需再次申请。）

AK/SK 认证方式

本节以身份证识别服务为例介绍如何以AK/SK认证方式使用SDK。

1. 获取AK/SK，具体步骤请参见[认证鉴权>AK/SK认证](#)。
2. 配置Node.js SDK的AK/SK。

根据获取的AK/SK，修改Demo工程“demo.js”文件中“appKey”和“appSecret”的值。

图 7-1 修改 demo.js 文件

```
// ak/sk
let appKey = 'xxx'; // your ak
let appSecret = 'xxx'; // your sk
let endPoint2 = 'ocr.cn-north-4.myhuaweicloud.com';
let httpUri2 = '/v1.0/ocr/id-card';
let ocrClientAksk = new OcrClientAksk(appKey, appSecret, endPoint2);
let opt2 = {};
//let opt2 = {'side': 'front'}; // option
hwOcrIdCard(ocrClientAksk, httpUri2, './data/id-card-demo.jpg',opt2);
```

3. 如使用本地图片文件进行识别，修改demo.js文件参数为本地图片路径。如使用SDK默认图片则不需要进行修改。
4. 执行demo.js文件，控制台输出200即表示程序执行成功。在控制台可以查看身份证识别的结果。

```
{
  "result": {
```

```
"name": "xx",
"sex": "女",
"ethnicity": "满",
"birth": "1990-xx-xx",
"address": "河北省承德市围场满族蒙古族自治县金车路",
"number": "3892011990012xxxxx"
}
}
```

📖 说明

如果想调用OCR其他识别服务，只需要把requestOcrServiceBase64函数的第一个参数改成对应的URI即可。URI列表请参见[表1 接口与API对应关系表](#)。

Token 认证方式

本节以身份证识别服务为例介绍如何以Token认证方式使用SDK。

1. 打开demo.js文件，修改main函数中的username、password和domainName为系统中实际注册的用户名、密码和域名（如果用户为非IAM用户，域名与用户名一致）。

图 7-2 修改 demo.js 文件中的用户名、密码和域名

```
// token
let domainName = 'xxx'; // if the user isn't IAM user, domainName is the same with username
let username = 'xxx';
let password = 'xxx';
let region = 'cn-north-4';
let httpUri1 = '/v1.0/ocr/id-card';
let hwOcrClientToken = new HwOcrClientToken(domainName,username, password, region);
let opt1 = {};
//let opt1 = {'side': 'front'}; // option
hwOcrIdCard(hwOcrClientToken, httpUri1, './data/id-card-demo.jpg', opt1);
```

2. 直接执行相应代码，在控制台可看到使用Token方式身份证识别服务的识别结果。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

A 修订记录

发布日期	修订说明
2020-04-30	修改 <ul style="list-style-type: none">• 使用SDK (Java)• 使用SDK (Python)• 使用SDK (iOS)• 使用SDK (Android)• 使用SDK (Node.js)
2019-02-02	修改 将文字识别对应Endpoint修改为“ocr.cn-north-1.myhuaweicloud.com”。
2018-12-12	新增 <ul style="list-style-type: none">• 文字识别服务SDK环境搭建，新增Python、iOS、Android、Node.js开发环境配置方法。• 使用文字识别SDK，新增Python、iOS、Android、Node.js示例。
2018-09-12	修改 申请服务操作步骤参见文档为《文字识别API参考》。
2018-07-30	修改 删除银行卡识别、一维码识别、二维码识别和车牌识别相关内容。

发布日期	修订说明
2018-05-10	新增 <ul style="list-style-type: none">• 银行卡识别服务的demo用例• 一维码识别服务的demo用例• 二维码识别服务的demo用例• 车牌识别服务的demo用例• 机动车销售发票识别的demo用例• 通用文字识别的demo用例
2017-12-30	第一次正式发布。